



User Guide

Amazon Lightsail



Amazon Lightsail: User Guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is Amazon Lightsail?	1
Features	1
Who is Lightsail for?	3
Access Lightsail	3
Get started	4
Related services	4
Estimates, billing, and cost optimization	5
Set up	6
Sign up for AWS	6
Create an IAM user	6
Get started	8
Step 1: Complete the prerequisites	8
Step 2: Create an instance	8
Step 3: Connect to your instance	10
Step 4: Add storage to your instance	11
Step 5: Create a snapshot	12
Step 6: Clean up	12
Next steps	13
Get started with Linux	13
Create a Linux-based instance	13
Connect to your instance	15
Next steps	17
Get started with Windows	18
Choose a Windows Server-based instance	18
Create a Windows Server-based instance	19
Connect to your instance	23
Instances	26
Create an instance	26
How to connect to your instance	28
Next steps	30
Delete an instance	31
Delete an instance from the Lightsail console home page	31
Delete an instance from the Lightsail console instance management page	32
Delete an instance using the AWS CLI	32

Next steps	34
Instance images	34
Compare platforms	35
Compare operating systems	35
Compare Database applications	39
Compare CMS applications	40
Compare application stacks and servers	42
Ecommerce applications	44
Project management applications	44
IPv6-only instance plans	45
What are IPv6-only instance plans	45
IPv6 considerations	45
Migrate to an IPv6-only instance	46
SSH key pairs	46
Choose a key pair option	47
Connect to your instances	48
Manage keys stored on instances	49
Connect to Linux instances	49
Connect to Windows instances	93
Instance snapshots	108
Connect to Linux EC2 instances	110
Connect to Windows EC2 instances	117
Windows snapshot and sysprep	124
Secure Windows EC2 instances	130
Secure Linux/Unix EC2 instances	132
Instance management	140
Start, stop, or restart your instance	141
Enhanced networking	143
Extend Windows storage	145
Linux shell scripts	149
PowerShell scripts	151
Windows security best practices	154
Instance firewall rules	157
Web server rules	158
Rules to connect to your instance from your computer	158
Database server rules	159

DNS server rules	160
SMTP email	160
Instance firewalls	160
Add and edit firewall rules	169
Instance Metadata Service	173
Use the Instance Metadata Service	173
Additional IMDS documentation	174
Configure IMDS	174
Disks	181
Block storage disks	181
Disk Quotas	182
Create and attach Linux/Unix disks	182
Step 1: Create a new disk and attach it to your instance	182
Step 2: Connect to your instance to format and mount the disk	184
Step 3: Mount the disk every time you reboot your instance	187
Create and attach Windows disks	188
Step 1: Create a new block storage disk and attach it to your instance	189
Step 2: Connect to your instance and bring the block storage disk online	191
Step 3: Initialize the block storage disk	193
Step 4: Format the disk with a file system	194
Detach and delete	196
Prerequisites	197
Detach and delete your disk	197
Snapshots	198
Manual snapshots	198
Automatic snapshots	199
System disk snapshots	199
Create new resources from snapshots	199
Copy snapshots	200
Export snapshots to Amazon EC2	200
Delete snapshots	200
Create snapshots	200
Create disk from snapshot	201
Create a root volume snapshot	205
Create an instance from a snapshot	215
Create a larger resource from a snapshot	218

Create a larger resource from a snapshot using the AWS CLI	220
Delete snapshots	225
Automatic snapshots	226
Automatic snapshot restrictions	227
Automatic snapshot retention	227
Enable or disable automatic instance snapshots using the Lightsail console	228
Enable or disable automatic snapshots for instances or block storage disks using the AWS CLI	229
Change snapshot time	233
Delete automatic snapshots	238
Keep automatic snapshots	242
Copy snapshots between Regions	248
Prerequisites	248
Copy a snapshot	248
Next steps	250
Export snapshots to EC2	251
Create Amazon EC2 resources from exported Lightsail snapshots	252
Choosing an Amazon EC2 instance type	254
Connect to Amazon EC2 instances	255
Secure an Amazon EC2 instance	255
Export Lightsail snapshots and create resources in Amazon EC2	256
How to export snapshots	256
Create EBS volumes from exported snapshots	261
Create EC2 instances from exported snapshots	263
Lightsail task monitor	274
Domains and DNS	276
How domain registration works	276
Domains that you can register in Lightsail	277
Pricing for domain registration	278
Additional information about domains	278
DNS in Lightsail	278
DNS terminology	279
DNS record types supported in the Lightsail DNS zone	281
Create a DNS zone	283
Edit or delete DNS zone	291
Internet traffic routing	292

Point domain to an instance	295
Point domain to a load balancer	297
Use another DNS service	300
Use Route 53	301
Register a domain	305
Register a new domain by using Lightsail	306
Domain details	309
Format domain names	310
Format domain names for domain name registration	311
Format domain names for DNS zones and records	311
Using an asterisk (*) in the names of DNS zones and records	311
Next steps	313
Manage domain in R53	313
View the status of a domain registration	313
Lock a domain to prevent unauthorized transfer to another registrar	314
Restore an expired or deleted domain	314
Transfer domain registrations	314
Delete a domain name registration	314
Registration information	314
Term	315
Automatic domain renewal	316
Registrant, administrative, and technical contacts	316
Same as registrant	316
Contact type	316
First name, last name	317
Organization	317
Email	317
Phone	317
Address 1	318
Address 2	318
Country	318
State	318
City	318
Postal/zip code	318
Privacy protection	318
Registration renewal	319

Automatic renewal	320
Configure automatic renewal for a domain during domain registration	321
Configure automatic renewal for a domain that is already registered	322
Privacy protection	322
Complete the prerequisites	323
Manage privacy protection for your domain	323
Domain contact information	323
Who is the owner of a domain?	324
Update contact information for a domain	324
Databases	325
Compare databases	325
Compare managed databases in Lightsail	325
Optimize data import	327
High availability databases	327
Create a database	328
Next steps	331
Connect to MySQL	331
Step 1: Get your MySQL database connection details	332
Step 2: Configure the public availability of your MySQL database	333
Step 3: Configure your database client to connect to your MySQL database	334
Next steps	336
Connect to MySQL using SSL	336
Supported connections	337
Prerequisites	337
Connect to your MySQL database using SSL	338
Connect to PostgreSQL	339
Step 1: Get your PostgreSQL database connection details	340
Step 2: Configure the public availability of your PostgreSQL database	341
Step 3: Configure your database client to connect to your PostgreSQL database	341
Next steps	344
Connect to PostgreSQL using SSL	344
Prerequisites	345
Connect to your Postgres database using SSL	345
Delete a database	346
Data import mode	347
Import data MySQL	348

Import data PostgreSQL	350
Database logs	352
MySQL query logs	354
Database snapshots	358
Next steps	359
Create database from backup	359
Create database from snapshot	362
Download SSL certificate	365
Certificate bundles for all AWS Regions	365
Certificate bundles for specific AWS Regions	365
Update CA certificate	366
Maintenance and backup windows	369
Prerequisites	369
Change your database maintenance window	370
Next steps	373
Manage database password	373
Next steps	375
Public mode	375
Next steps	376
Update parameters	376
Prerequisites	377
Get a list of available database parameters	377
Update your database parameters	379
Upgrade major version	380
Prerequisites	381
Update the database major version	381
Next steps	385
Load balancers	386
Load balancer features	386
When to use load balancers	387
Recommended applications for load balancing	387
Get started using load balancers	387
Create load balancer	388
Prerequisites	388
Create a load balancer	388
Attach an instance to your load balancer	389

Next steps	390
Load balancer SSL/TLS certificates	390
Prerequisites	390
Create the certificate request	391
Next step	391
Add alternate domains	392
Verify certificate	393
Attach certificate to load balancer	399
Delete certificate	399
Update load balancer settings	400
Health checks	400
Encrypted traffic (HTTPS)	401
Session persistence	401
Instance load balancing	402
General guidelines: Applications that use a database	402
WordPress	402
Node.js	402
Magento	403
GitLab	403
Drupal	404
LAMP stack	404
MEAN stack	405
Redmine	405
Nginx	405
Joomla!	405
Configure TLS security policy	406
Security policies overview	406
Supported security policies and protocols	407
Complete the prerequisites	409
Configure a security policy using the Lightsail console	409
Configure a security policy using the AWS CLI	409
HTTP to HTTPS redirect	410
Complete the prerequisites	411
Configure HTTPS redirection on your load balancer using the Lightsail console	411
Configure HTTP to HTTPS redirect for a load balancer with the AWS CLI	412
Session persistence	413

Enable session persistence	413
Adjust the cookie duration	414
Health checks	415
Customize your health check path	415
Health check metrics	416
Health check status	418
Detach instances	419
Delete load balancer	419
Distributions	420
Use cases	422
Configure your distribution	423
Edge locations and IP address ranges	425
Create a distribution	425
Prerequisites	425
Origin resource	426
Origin protocol policy	427
Caching behavior and caching presets	428
Best for WordPress caching preset	429
Default behavior	429
Directory and file overrides	430
Advanced cache settings	431
Distribution plan	435
Create a distribution	435
Next steps	438
Delete a distribution	439
Delete your distribution	439
Caching behavior	439
Caching preset	440
Best for WordPress caching preset	440
Default behavior	441
Directory and file overrides	441
Advanced cache settings	442
Change your distribution's cache behavior	445
Reset cache	446
Change origin	447
Origin protocol policy	447

Change your distribution's origin	448
Change plan	449
Change your distribution plan	450
Distribution custom domains	450
Prerequisites	451
Enable custom domains for your distribution	451
Point your domain to a distribution	452
Change custom domain	454
Disable distribution custom domains	455
Add distribution domain to container service	456
Request and response behaviors	458
How your distribution processes and forwards requests to your origin	458
How your distribution processes responses from your origin	473
Test distribution	478
Test your distribution	478
Networking	480
Load balancers	480
Static IPs	480
Regions and Availability Zones	480
SSH keys and Lightsail regions	481
Tips for working with Lightsail regions	481
Lightsail Availability Zones	482
Availability Zones and your Lightsail application	482
Configure reverse DNS	482
Prerequisites	483
Submit a request to AWS Support to configure reverse DNS	484
VPC peering	485
IP addresses	486
Private and public IPv4 addresses for instances	487
Static IPv4 addresses for instances	488
IPv6 for instances, container services, CDN distributions, and load balancers	490
Static IP addresses	492
Enable or disable IPv6	497
SSL/TLS Certificates	501
Why use HTTPS?	502
Process overview	502

Use SSL/TLS certificates with your distribution or container service	503
Use SSL/TLS certificates with your load balancer	504
Container certificates	504
Distribution certificates	510
Buckets	522
Object storage concepts	522
Manage buckets and objects	524
Create buckets	525
Create a bucket	525
Manage buckets and objects	526
Delete buckets	528
Force deleting a bucket	528
Delete your bucket using the Lightsail console	529
Delete your bucket using the AWS CLI	529
Manage buckets and objects	531
Access keys	533
Create access keys for a bucket	534
Block public access	535
Configuring block public access settings for your account	535
Manage buckets and objects	538
Bucket access logs	540
What do I need to enable log delivery?	541
Log object key format	541
How are logs delivered?	542
Best effort access log delivery	542
Bucket logging status changes take effect over time	542
Access log format	543
Enable access logs	556
Use access logs	560
Bucket objects	565
Filter objects using the Lightsail console	565
View objects using the AWS CLI	567
Manage buckets and objects	569
Copy and move objects	571
Delete objects	576
Download objects	584

Filter objects	588
Manage object versioning	592
Restore object versions	597
Tag objects	601
Bucket resource access	606
Configure resource access for a bucket	606
Change bucket plans	607
Change your bucket's storage plan using the Lightsail console	607
Change your bucket's storage plan using the AWS CLI	608
Configure access permissions	609
Configure bucket access permissions	610
Cross-account access	611
Configure cross-account access for a bucket	612
Individual object access permissions	612
Configure individual object access permissions	613
Multipart upload	614
Multipart upload process	615
Concurrent multipart upload operations	617
Multipart upload retention	618
Amazon Simple Storage Service multipart upload limits	618
Split the file to upload	619
Initiate a multipart upload using the AWS CLI	619
Upload a part using the AWS CLI	620
List parts of a multipart upload using the AWS CLI	621
Create a multipart upload .json file	623
Complete a multipart upload using the AWS CLI	625
List multipart uploads for a bucket using the AWS CLI	626
Stop a multipart upload using the AWS CLI	627
Naming rules	628
Example bucket names	629
Object key names	629
Key names	630
Object key naming guidelines	630
XML related object key constraints	633
Object storage security best practices	633
Preventative security best practices	634

Monitoring and auditing best practices	639
Understanding bucket permissions	640
Bucket access permissions	641
Individual object access permissions	642
Cross-account access	642
Access keys	642
Resource access	643
Amazon S3 block public access	643
Upload files to bucket	643
Object key names and versioning	644
Upload files to a bucket using the Lightsail console	644
Upload files to a bucket using the AWS CLI	645
Configure the AWS CLI for IPv6-only requests	646
Managing buckets and objects in Lightsail	647
Container services	650
Containers	651
Lightsail container service elements	651
Lightsail container services	651
Container service capacity (scale and power)	652
Pricing	653
Deployments	653
Deployment versions	654
Container image sources	655
Public endpoints and default domains	655
Custom domains and SSL/TLS certificates	656
Container logs	657
Metrics	657
Use Lightsail container services	657
Create a container	659
Container service capacity (scale and power)	659
Pricing	660
Container service status	660
Create a container service	661
Delete a container	664
Delete a container service	664
Container images	664

Step 1: Complete the prerequisites	665
Step 2: Create a Dockerfile and build a container image	665
Step 3: Run your new container image	667
(Optional) Step 4: Clean up the containers running on your local machine	668
Next steps after creating container images	669
Manage container images	669
Install the plugin	674
ECR private repository access	681
Manage containers and deployments	699
Prerequisites	700
Deployment parameters	701
Communication between containers	705
Container logs	705
Deployment versions	706
Deployment status	706
Deployment failures	706
View your current container service deployment	707
Create or modify your container service deployment	707
Change container capacity	709
Manage deployment versions	710
View container logs	712
Container service custom domains	714
Container service custom domain limits	715
Prerequisites	716
View custom domains for a container service	716
Enable custom domains for a container service	717
Disable custom domains for a container service	718
Point Lightsail domain to container	718
Point Route 53 domain to container	721
Security	727
Infrastructure security	727
Resilience	728
Identity and access management	728
Audience	728
Authenticating With Identities	729
Managing Access Using Policies	733

AWS managed policies	737
Lightsail policies and roles	739
Manage IAM user access	761
Update management	767
Instance blueprint software support	768
Compliance validation	769
Monitor resources	770
Monitoring your resources effectively	770
Metric concepts and terminology	771
Metrics	771
Metrics retention	771
Statistics	771
Units	772
Periods	772
Alarms	773
Metrics available in Lightsail	773
Instance metrics	773
Database metrics	774
Distribution metrics	775
Load balancer metrics	775
Container service metrics	777
Bucket metrics	777
Resource health metrics	777
Instance metrics	778
Database metrics	779
Distribution metrics	779
Load balancer metrics	780
Container service metrics	781
Bucket metrics	782
Metric notifications	782
Instance burst capacity	783
View instance metrics	793
Metric alarms	797
Create instance alarms	808
Delete or disable alarms	813
Bucket metrics	814

Bucket metrics	815
View bucket metrics in the Lightsail console	815
Manage buckets and objects	816
Create alarms	818
Container metrics	822
Container service metrics	822
View container service metrics in the Lightsail console	823
Database metrics	824
Database metrics	824
Viewing database metrics in the Lightsail console	825
Next steps after viewing your database metrics	825
Create database alarms	826
Distribution metrics	831
Distribution metrics	832
View distribution metrics in the Lightsail console	832
Next steps after viewing your distribution metrics	833
Create distribution alarms	833
Load balancer metrics	839
Load balancer metrics	839
View load balancer metrics	840
Next steps	841
Load balancer alarms	841
Add notification contacts	847
Regional notification contact limits	848
SMS text messaging support	848
Email contact verification	849
Adding notification contacts using the Lightsail console	849
Adding notification contacts using the AWS CLI	854
Next steps after adding your notification contacts	856
Delete notification contacts	856
Deleting notification contacts using the Lightsail console	857
Deleting notification contacts using the AWS CLI	857
Next steps after deleting your notification contacts	858
Tags	859
Use tags to organize billing and control access	859
Lightsail resources that support tagging	860

Tag restrictions	861
Add tags	861
Next steps	863
Delete tags	864
Permissions and authorization based on tags	866
Use tags to control access	866
Step 1: Create an IAM policy	866
Step 2: Attach the policy to users or groups	868
Use tags to organize costs	868
Step 1: Add key-value tags to resources	868
Step 2: Activate user-defined cost allocation tags	869
Step 3: Set up the cost allocation report, and view it	869
Use tags to organize resources	870
View tags for a resource	870
Filter resources using tags	871
Troubleshooting	874
WordPress setup	874
Common errors	875
Setup failures	878
403 error (unauthorized)	881
Block storage disks	882
General disk errors	882
Browser-based SSH or RDP client	884
Error message: Can't connect	884
Error message: Can't connect right now	886
Ghost service unavailable	887
Start the Ghost service	887
IAM issues	890
I am not authorized to perform an action in Lightsail	890
I am not authorized to perform iam:PassRole	891
I want to view my access keys	891
I'm an administrator and want to allow others to access Lightsail	892
I want to allow people outside of my AWS account to access my Lightsail resources	892
IPv6 reachability	893
Enable IPv6 for dual-stack instances	893
Configure the instance's firewall	894

Test reachability to your instance	895
Insufficient instance capacity error	898
Insufficient capacity when launching a new instance	898
Insufficient capacity when starting a stopped instance	899
Related information	899
Load balancers	899
General load balancer errors	899
Notifications	900
SSL/TLS certificates	902
Tutorials	903
Quick start guides	903
cPanel & WHM	904
Drupal	917
Ghost	927
GitLab CE	940
Joomla!	953
LAMP	966
Magento	968
Nginx	986
Node.js	988
Plesk	990
PrestaShop	993
Redmine	1009
WordPress	1019
WordPress Multisite	1026
Bitnami	1036
Bitnami user name and password	1036
Remove Bitnami banner	1043
WordPress	1046
Configure WordPress	1046
Connect to Amazon S3	1055
Connect to Aurora DB	1063
Connect to MySQL	1071
Connect to a storage bucket	1075
Configure a CDN	1091
Enable email	1095

Enable HTTPS	1105
Migrate to Lightsail	1116
WordPress Multisite	1124
WordPress Multisite: Add blogs as domains	1124
WordPress Multisite: Add blogs as subdomains	1131
WordPress Multisite: Define domain	1135
Let's Encrypt	1137
LAMP Let's Encrypt certificate	1137
Nginx Let's Encrypt certificate	1152
WordPress Let's Encrypt certificate	1169
Networking	1185
IPv6 for cPanel and WHM	1186
IPv6 for Debian 8	1192
IPv6 for GitLab	1196
IPv6 for Nginx	1199
IPv6 for Plesk	1202
IPv6 for Ubuntu 16	1205
Working with Lightsail	1209
AWS CLI for Lightsail	1210
Set up access keys	1210
AWS CloudShell	1212
CloudTrail logging	1216
Connect a LAMP instance to an Aurora database	1218
Create a HAR file	1223
Force stop an instance	1226
Install Prometheus on a Linux-based instance	1228
Launch and configure LAMP	1242
Launch and configure Windows Server 2016	1251
Learn more about Lightsail	1259
Migrate from a MySQL 5.6 database	1266
Set up Plesk	1274
Use buckets with distributions	1280
Work with other AWS services	1300
AWS CloudFormation resources	1309
Billing	1313
View your detailed Lightsail bill	1313

Billing usage types	1314
Region codes in your bill	1316
FAQs	1317
General	1317
Instances	1320
Object storage and buckets	1323
Container services	1326
Databases	1329
Block storage	1333
Load balancers	1335
Content delivery network distributions	1337
Certificates	1341
Manual and automatic snapshots	1342
Networking	1344
Domains	1346
Billing and account management	1347
Export to Amazon Elastic Compute Cloud (Amazon EC2)	1353
Tags in Lightsail	1354
Contacts and notifications	1356
Metrics and alarms	1356
Get help	1358
Context-sensitive help panel	1358
About this user guide	1358
Using search	1359
Using the Lightsail CLI and API	1359
AWS forums and other community resources	1359

What is Amazon Lightsail?

Amazon Lightsail is the easiest way to get started with Amazon Web Services (AWS) for anyone who needs to build websites or web applications. It includes everything you need to launch your project quickly—instances (virtual private servers), container services, managed databases, content delivery network (CDN) distributions, load balancers, SSD-based block storage, static IP addresses, DNS management of registered domains, and resource snapshots (backups)—for a low, predictable monthly price.

Lightsail also offers Amazon Lightsail for Research. With Lightsail for Research, academics and researchers can create powerful virtual computers in the AWS Cloud. These virtual computers come with pre-installed research applications, such as RStudio and Scilab. For more information see the [Amazon Lightsail for Research User Guide](#).

Topics

- [Features of Lightsail](#)
- [Who is Lightsail for?](#)
- [Access Lightsail](#)
- [Get started with Lightsail](#)
- [Related services](#)
- [Estimates, billing, and cost optimization](#)

Features of Lightsail

Lightsail provides the following high-level features:

Instances

Lightsail offers virtual private servers (instances) that are easy to set up and backed by the power and reliability of AWS. You can launch your website, web application, or project in minutes, and manage your instance from the intuitive Lightsail console or API.

As you're creating your instance, you'll click-to-launch a simple operating system (OS), a pre-configured application, or development stack—such as WordPress, Windows, Plesk, LAMP, Nginx, and more. Every Lightsail instance comes with a built-in firewall that you can use to allow or restrict traffic to your instances based on source IP, port, and protocol. [Learn more](#)

Containers

Run and securely access containerized applications in the cloud. A container is a standard unit of software that packages code and its dependencies together so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Load balancers

Route web traffic across your instances so your websites and applications can accommodate variations in traffic, protected against outages, and deliver a seamless visitor experience. [Learn more](#)

Managed databases

Lightsail offers a fully configured MySQL or PostgreSQL databases plan that includes memory, processing, storage, and transfer allowance. With Lightsail managed databases, you can easily scale your databases independently of your virtual servers, improve application availability, or run standalone databases in the cloud. [Learn more](#)

Block and object storage

Lightsail offers both block and object storage. You can scale your storage quickly and easily with highly available SSD-backed storage for your Linux or Windows virtual server. [Learn more](#)

With Lightsail Object storage buckets, you can store and retrieve objects, at any time, from anywhere on the internet. You can also host static content on the cloud. [Learn more](#)

CDN distributions

Lightsail enables content delivery network (CDN) distributions, which are built on the same infrastructure as Amazon CloudFront. You can easily distribute your content to a global audience by setting up proxy servers across the world, so that your users can access your website geographically closer to them, thus reducing latency. [Learn more](#)

Access to AWS services

Lightsail uses a focused set of features like instances, managed databases and load balancers to make it easier to get started. But that doesn't mean you're limited to those options –you can integrate your Lightsail project with some of the 90+ other services in AWS through Amazon VPC peering. [Learn more](#)

For more details about Lightsail, see [Amazon Lightsail](#).

Who is Lightsail for?

Lightsail is for everyone. You can choose an image for your Lightsail instance that jump starts your project so you don't have to spend as much time installing software or frameworks.

If you're an individual developer or hobbyist working on a personal project, Lightsail can help you deploy and manage basic cloud resources. You might also be interested in learning or experimenting with cloud services, such as virtual machines, domains or networking. Lightsail provides a quick way to get started.

Lightsail has images with base operating systems, development stacks like LAMP, LEMP (Nginx), and SQL Server Express, and applications like WordPress, Drupal, and Magento. For more detailed information about the software installed on each image, see [Choose a Lightsail instance image](#).

As your project grows, you can add block storage disks and attach them to your Lightsail instance. You can take snapshots of these instances and disks and easily create new instances from those snapshots. You can also peer your VPC so that your Lightsail instances can use other AWS resources outside of Lightsail.

You can also create a Lightsail load balancer and attach target instances to create a highly available application. You can also configure your load balancer to handle encrypted (HTTPS) traffic, session persistence, health checking, and more.

Access Lightsail

You can create and manage your Lightsail resources with the following interfaces:

Amazon Lightsail console

A simple web interface to create and manage Lightsail instances and resources. If you've signed up for an AWS account, you can access the Lightsail console by signing into the AWS Management Console and selecting **Lightsail** from the console home page.

AWS Command Line Interface

Enables you to interact with AWS services using commands in your command-line shell. It is supported on Windows, Mac, and Linux. For more information about the AWS CLI, see [AWS Command Line Interface User Guide](#). You can find the Lightsail commands in the [Amazon Lightsail API Reference](#).

AWS Tools for PowerShell

A set of PowerShell modules that are built on the functionality exposed by the AWS SDK for .NET. The Tools for PowerShell enable you to script operations on your AWS resources from the PowerShell command line. To get started, see the [AWS Tools for Windows PowerShell User Guide](#). You can find the cmdlets for Lightsail, in the [AWS Tools for PowerShell Cmdlet Reference](#).

Query API

Lightsail provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named `Action`. For more information about the API actions for Lightsail, see [Actions](#) in the *Amazon Lightsail API Reference*.

AWS SDKs

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, AWS provides libraries, sample code, tutorials, and other resources for software developers. These libraries provide basic functions that automate tasks such as cryptographically signing your requests, retrying requests, and handling error responses, making it easier for you to get started. For more information, see [Tools to Build on AWS](#).

Get started with Lightsail

After you set up to use Lightsail, you can walk through [Tutorial: Get started with Amazon Lightsail instances](#) to launch, connect to, and clean up an instance.

Related services

You can provision Lightsail resources, such as instances and disks, directly using Lightsail. In addition, you can provision resources using other AWS services, such as the following:

- [Amazon EC2](#)

Provides resizable computing capacity—literally, servers in Amazon's data centers—that you use to build and host your software systems. To compare Lightsail and Amazon EC2, see [Amazon Lightsail or Amazon EC2](#).

- [Amazon EC2 Auto Scaling](#)

Helps ensure you have the correct number of Amazon EC2 instances available to handle the load for your application.

- [Elastic Load Balancing](#)

Automatically distribute incoming application traffic across multiple instances.

- [Amazon Relational Database Service \(Amazon RDS\)](#)

Set up, operate, and scale a managed relational database in the cloud.

- [Amazon Elastic Container Service \(Amazon ECS\)](#)

Deploy, manage, and scale containerized applications on a cluster of Amazon EC2 instances.

Estimates, billing, and cost optimization

To create estimates for your AWS use cases, use the [AWS Pricing Calculator](#).

To see your bill, go to the **Billing and Cost Management Dashboard** in the [AWS Billing and Cost Management console](#). Your bill contains links to usage reports that provide details about your bill.

To learn more about AWS account billing, see [AWS Billing and Cost Management User Guide](#).

If you have questions concerning AWS billing, accounts, and events, [contact AWS Support](#).

You can optimize the cost, security, and performance of your AWS environment using [AWS Trusted Advisor](#).

Set up your AWS account to use Amazon Lightsail

If you're a new AWS customer, complete the setup prerequisites that are listed on this page before you start using Amazon Lightsail. For these setup procedures, you use the AWS Identity and Access Management (IAM) service. For complete information about IAM, see the [IAM User Guide](#).

Topics

- [Sign up for AWS](#)
- [Create an IAM user](#)

Sign up for AWS

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, [assign administrative access to an administrative user](#), and use only the root user to perform [tasks that require root user access](#).

Create an IAM user

To create an administrator user, choose one of the following options.

Choose one way to manage your administrator	To	By	You can also
In IAM Identity Center (Recommended)	Use short-term credentials to access AWS. This aligns with the security best practices . For information about best practices , see Security best practices in IAM in the <i>IAM User Guide</i> .	Following the instructions in Getting started in the <i>AWS IAM Identity Center User Guide</i> .	Configure programmatic access by Configuring the AWS CLI to use AWS IAM Identity Center in the <i>AWS Command Line Interface User Guide</i> .
In IAM (Not recommended)	Use long-term credentials to access AWS.	Following the instructions in Creating your first IAM admin user and user group in the <i>IAM User Guide</i> .	Configure programmatic access by Managing access keys for IAM users in the <i>IAM User Guide</i> .

Tutorial: Get started with Amazon Lightsail instances

Use this tutorial to learn how to create, connect to, and use an Amazon Lightsail instance. In Lightsail, an instance is a virtual private server (also called a virtual machine). You create and manage Lightsail instances in the AWS Cloud. When you create an instance, you choose an image that has an operating system (OS) on it. You can also choose an instance image that has an application or development stack on it, including the base OS.

The instance that you create in this tutorial will incur usage fees from the time that you create the instance until you delete it. Deletion is the final step of this tutorial. For more information about pricing, see [Lightsail pricing](#).

Topics

- [Step 1: Complete the prerequisites](#)
- [Step 2: Create an instance](#)
- [Step 3: Connect to your instance](#)
- [Step 4: Add storage to your instance](#)
- [Step 5: Create a snapshot](#)
- [Step 6: Clean up](#)
- [Next steps](#)
- [Get started with Linux/Unix-based instances in Amazon Lightsail](#)
- [Get started with Windows Server-based instances in Amazon Lightsail](#)

Step 1: Complete the prerequisites

If you're a new AWS customer, complete the setup prerequisites before you start using Amazon Lightsail. For more information, see [Set up your AWS account to use Amazon Lightsail](#).

Step 2: Create an instance

You can create an instance by using the [Lightsail console](#) as described in the following procedure. This tutorial is intended to help you quickly launch your first instance. We also recommend exploring the available applications and hardware plans. For more information, see [Choose an Amazon Lightsail instance image](#).

1. Sign in to the [Lightsail console](#).
2. On the home page, choose **Create instance**.
3. Select a location for your instance (an AWS Region and Availability Zone). Choose an AWS Region that is closest to your physical location for reduced latency.

Choose **Change AWS Region and Availability Zone** to create your instance in another location.

4. You can pick an application (**Apps + OS**) or an operating system (**OS Only**).

To learn more about Lightsail instance images, see [Choose an Amazon Lightsail instance image](#).

5. Choose your instance plan.

Choose whether your instance uses dual-stack (IPv4 and IPv6), or IPv6-only networking. Some Lightsail blueprints don't support IPv6-only networking at this time. To see which blueprints support IPv6-only networking see [Choose an Amazon Lightsail instance image](#).

You can try the \$3.50 USD Lightsail plan free for one month (up to 750 hours). We will credit one free month to your account. Learn more on our [Lightsail pricing page](#).

6. Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

7. Choose **Create instance**.

Within minutes, your Lightsail instance is ready and you can connect to it.

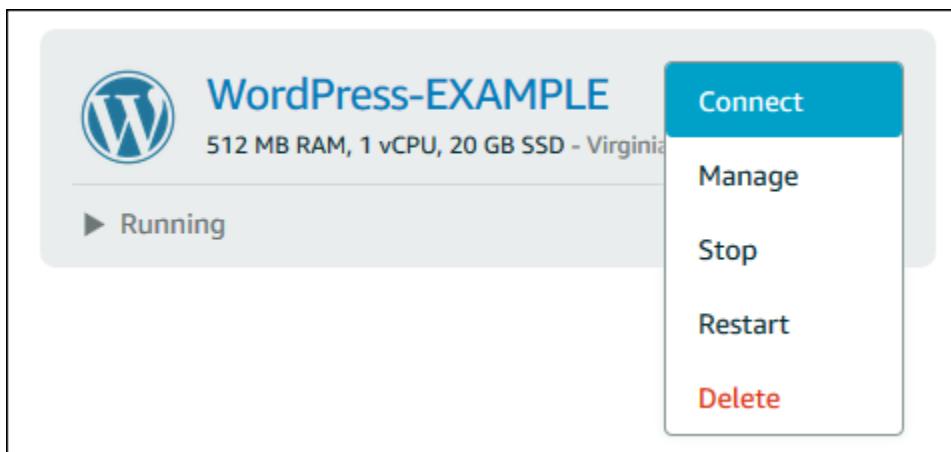
Step 3: Connect to your instance

1.

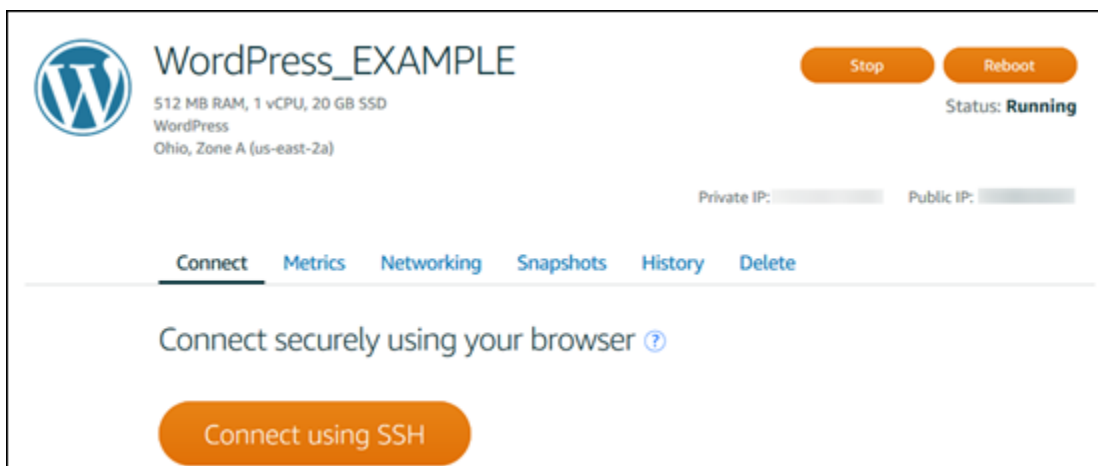
Note

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#)

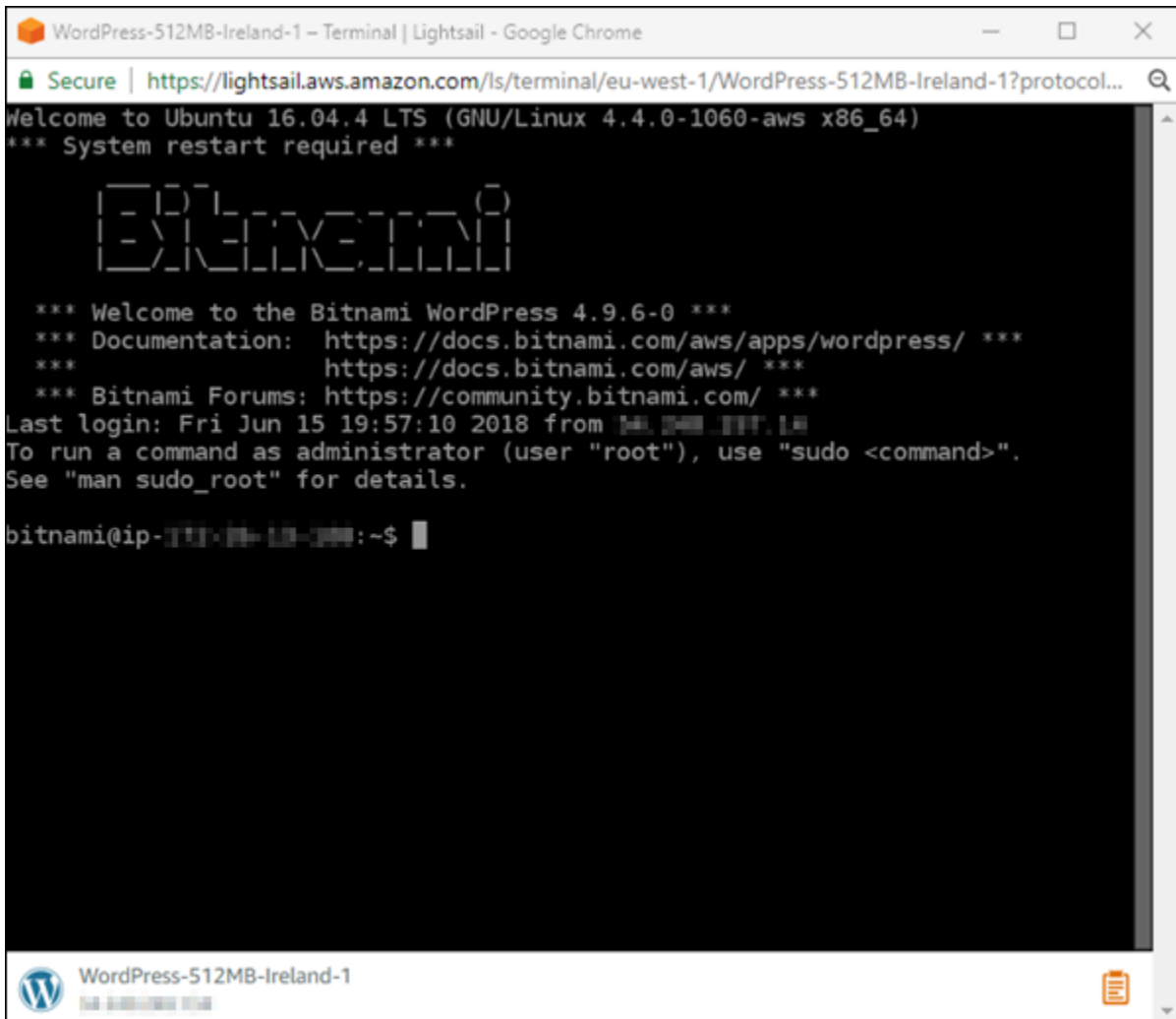
From the Lightsail home page, choose the menu on the right of your instance's name, and then choose **Connect**.



Alternately, you can open your instance management page and choose the **Connect** tab.



2. You can now type commands into the terminal and manage your Lightsail instance without setting up an SSH client.

A screenshot of a terminal window titled "WordPress-512MB-Ireland-1 - Terminal | Lightsail - Google Chrome". The terminal output shows a welcome message for Ubuntu 16.04.4 LTS, a system restart notice, the Bitnami logo, and a welcome message for Bitnami WordPress 4.9.6-0. It includes documentation and forum links, a last login timestamp, and instructions for running commands as administrator. The prompt is "bitnami@ip-10.130.10.11: ~\$".

```
WordPress-512MB-Ireland-1 - Terminal | Lightsail - Google Chrome
Secure | https://lightsail.aws.amazon.com/lightsail/terminal/eu-west-1/WordPress-512MB-Ireland-1?protocol...
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)
*** System restart required ***

  _ _ _ _ _
 / _ _ _ _ \\
/_ _ _ _ _ \\

*** Welcome to the Bitnami WordPress 4.9.6-0 ***
*** Documentation: https://docs.bitnami.com/aws/apps/wordpress/ ***
*** https://docs.bitnami.com/aws/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***
Last login: Fri Jun 15 19:57:10 2018 from 10.130.10.11
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bitnami@ip-10.130.10.11: ~$
```

To learn how to connect to add additional storage to your virtual computer, continue to the next step of this tutorial.

Step 4: Add storage to your instance

Lightsail provides block-level storage volumes (disks) that you can attach to an instance. Even though your instance comes with a system disk, you can attach additional storage disks as your needs change. You can also detach a disk from an instance and attach it to another instance.

After you create an additional disk, you'll need to connect to your Lightsail instance to format and mount the disk.

For more information about creating, attaching, and managing a disk, see [Create and attach Lightsail block storage disks to your Linux-based instance](#).

To learn about backing up your virtual computer, continue to the next step of this tutorial.

Step 5: Create a snapshot

Snapshots are a point-in-time copy of your data. You can create snapshots of your instances and use them as baselines to create new instances or for data backup. A snapshot contains all of the data that's needed to restore your instance (from the moment when the snapshot was taken).

For more information about creating and managing snapshots, see [Create a snapshot of your Linux or Unix Lightsail instance](#).

To learn about cleaning up your virtual computer resources, continue to the next step of this tutorial.

Step 6: Clean up

After you're done with the instance that you created for this tutorial, you can delete it. This stops incurring charges for the instance if you don't need it.

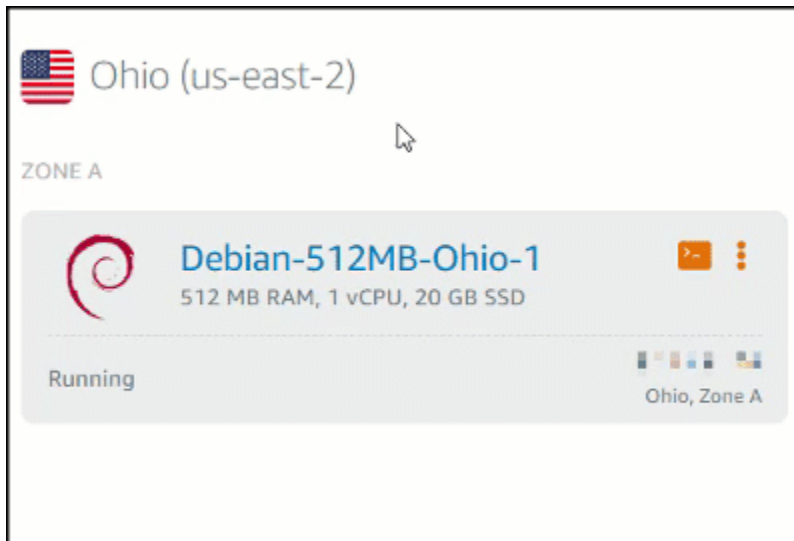
Deleting an instance doesn't delete its associated snapshots or attached disks. If you created snapshots and disks for this tutorial, you should delete those as well.

To save your instance for later, but to avoid incurring charges, you can stop the instance instead of deleting it. Then you can start it again later. For more information about pricing, see [Lightsail pricing](#).

Important

Deleting a Lightsail resource is a permanent action. The deleted data cannot be recovered. If you might need the data later, create a snapshot of your virtual computer before you delete it. For more information, see [Create a snapshot of your Linux or Unix Lightsail instance](#).

1. Sign in to the [Lightsail console](#).
2. Choose **Instances** in the navigation pane.
3. For the instance you want to delete, choose the actions menu icon (:), then choose **Delete**.



4. Choose **Yes, delete** to confirm the deletion.

Next steps

Use the following topics to get started with Amazon Lightsail Linux and Windows based instances.

- [Get started with Linux/Unix-based instances in Amazon Lightsail](#)
- [Get started with Windows Server-based instances in Amazon Lightsail](#)

Get started with Linux/Unix-based instances in Amazon Lightsail

You can create a Linux/Unix-based Lightsail instance (a virtual private server) running an application like WordPress or a development stack like LAMP in seconds. After your instance starts running, you can connect to it via SSH without leaving Lightsail. Here's how.

To create a Windows-based instance, see [Get started with Windows-based instances in Amazon Lightsail](#).

Create a Linux-based instance

1. On the home page, choose **Create instance**.
2. Select a location for your instance (an AWS Region and Availability Zone).

Choose **Change AWS Region and Availability Zone** to create your instance in another location.

3. Optionally, you can change the Availability Zone.

Choose **Change your Availability Zone**.


4. Choose the Linux platform.
5. Pick an application (**Apps + OS**) or an operating system (**OS Only**).

To learn more about Lightsail instance images, see [Choose an Amazon Lightsail instance image](#).

6. Choose your instance plan.

Choose whether your instance uses dual-stack (IPv4 and IPv6), or IPv6-only networking. Some Lightsail blueprints don't support IPv6-only networking at this time. To see which blueprints support IPv6-only networking see [Choose an Amazon Lightsail instance image](#).

You can try the \$3.50 USD Lightsail plan free for one month (up to 750 hours). We'll credit one free month to your account. Learn more on our [Lightsail pricing page](#).

 **Note**

As part of the AWS Free Tier, you can get started with Amazon Lightsail for free on select instance bundles. For more information, see **AWS Free Tier** on the [Amazon Lightsail Pricing page](#).

7. Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

8. Choose one of the following options to add tags to your instance:

- **Add key-only tags.** Enter your new tag into the tag key text box, and press **Enter**. Choose **X** to remove any tags you don't want to keep.

- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Key-value tags can only be added one at a time. Choose **Add key-value tag** to add additional key-value tags, or choose **X** to remove any tags you don't want to keep.

Note

For more information about key-only and key-value tags, see [Tags](#).

9. Choose **Create instance**.

For advanced creation options, see [Use a launch script to configure your Amazon Lightsail instance when it starts up](#) or [Set up SSH for your Linux/Unix-based Lightsail instances](#).

Within minutes, your Lightsail instance is ready and you can connect to it via SSH, without leaving Lightsail!

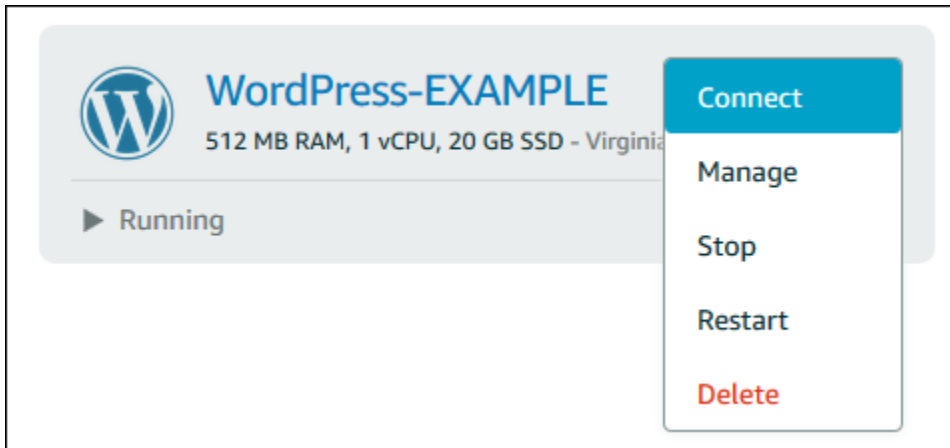
Connect to your instance

- 1.

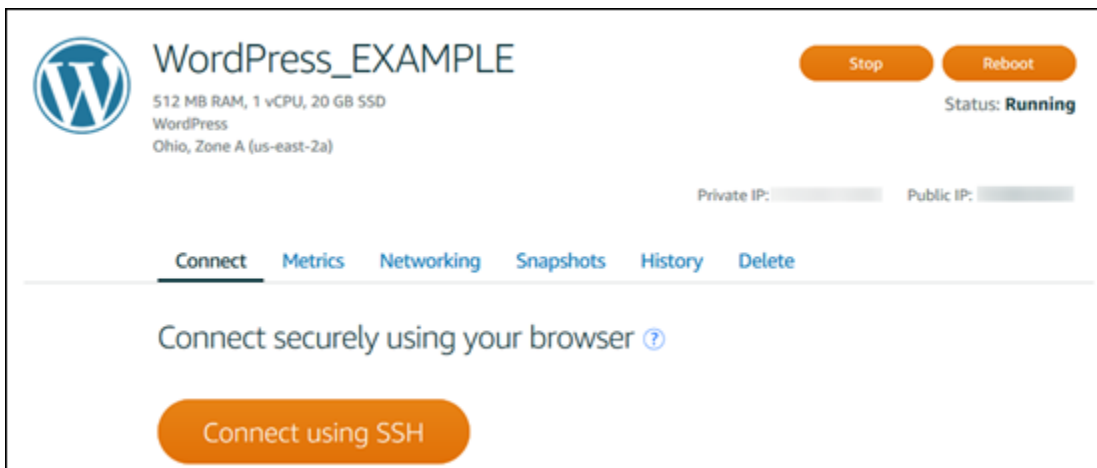
Note

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#)

On the Lightsail home page, choose the menu on the right of your instance's name, and then choose **Connect**.



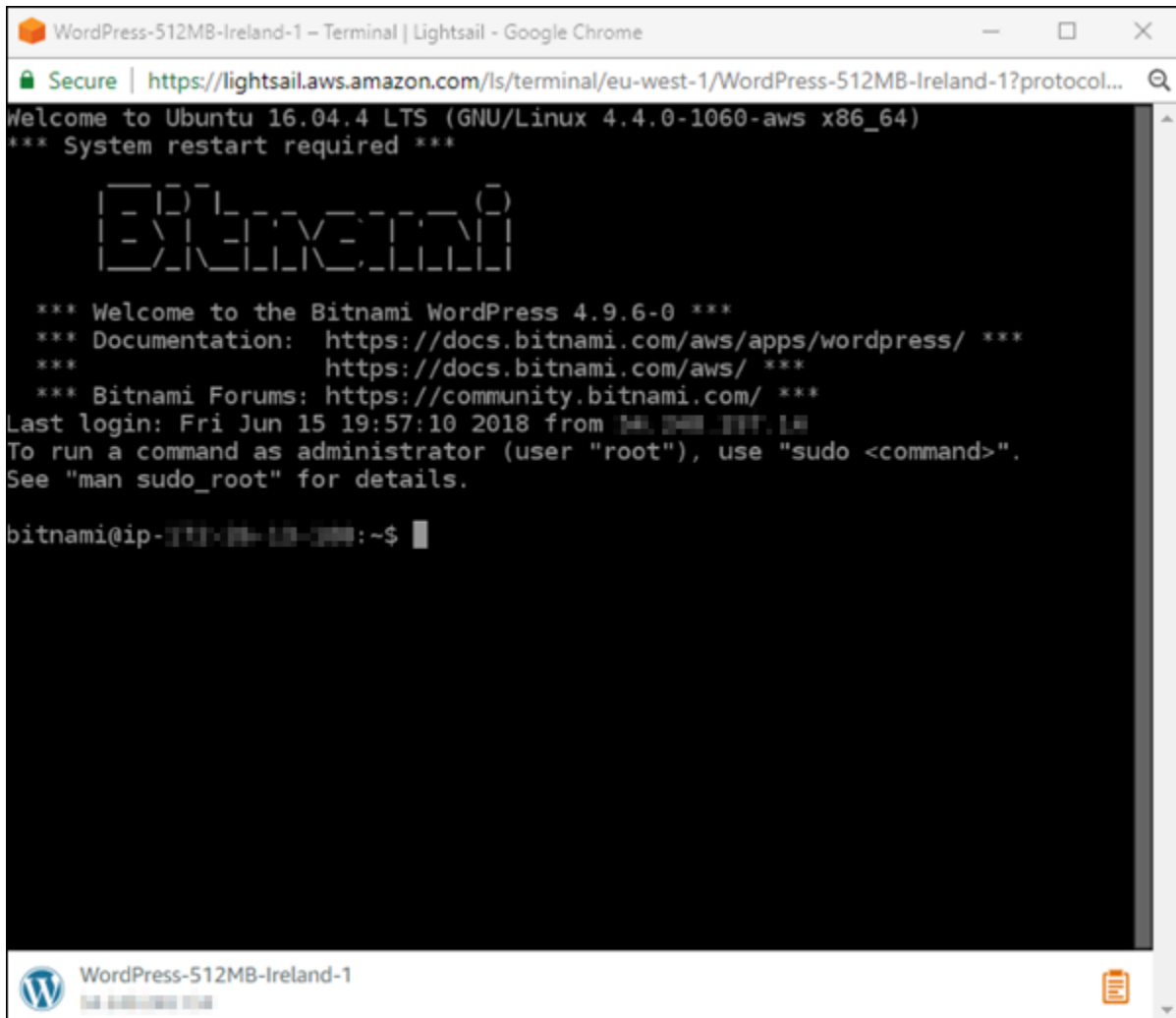
Alternately, you can open your instance management page and choose the **Connect** tab.



Note

To connect to your instance using an SSH client such as PuTTY, you can follow this procedure: [Set up PuTTY to connect to your Lightsail instance.](#)

2. Now you can type commands into the terminal and manage your Lightsail instance without setting up an SSH client.



```
WordPress-512MB-Ireland-1 – Terminal | Lightsail - Google Chrome
Secure | https://lightsail.aws.amazon.com/ls/terminal/eu-west-1/WordPress-512MB-Ireland-1?protocol...
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)
*** System restart required ***

          _ _ _
         | | |
        _||_|
       /___\

*** Welcome to the Bitnami WordPress 4.9.6-0 ***
*** Documentation: https://docs.bitnami.com/aws/apps/wordpress/ ***
***               https://docs.bitnami.com/aws/          ***
*** Bitnami Forums: https://community.bitnami.com/      ***
Last login: Fri Jun 15 19:57:10 2018 from 10.132.1.10
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bitnami@ip-10.132.1.10:~$
```

Next steps

Now that you can connect to your instance, what you do next depends on how you plan to use it. For example:

- [the section called “WordPress”](#) if you're creating a blog.
- [Create a static IP address](#) for your instance to keep the same IP address each time you restart your Lightsail instance.
- [Create a snapshot of your instance](#) as a backup.

Get started with Windows Server-based instances in Amazon Lightsail

You can create Lightsail instances that run the Windows Server operating system (OS). We have three OS blueprints available: Windows Server 2022, Windows Server 2019, and Windows Server 2016. In addition, we have blueprints that come preconfigured with SQL Server 2022, 2019, and 2016 Express.

This topic provides information about choosing your software, creating your Windows Server-based instance, and connecting to it.

Learn more about [Windows Server on AWS](#)

Choose a Windows Server-based instance

There are three options for creating a Windows Server-based instance in Lightsail.

Windows Server 2022

Lightsail running Windows Server is a fast and dependable environment for deploying applications using the Microsoft Web Platform. With Lightsail, you can run any compatible Windows-based solution on the high-performance, reliable, cost-effective AWS Cloud computing platform. Common Windows use cases include Enterprise Windows-based application hosting, website and web service hosting, data processing, distributed testing, ASP.NET application hosting, and any other application requiring Windows software.

[Learn more about the Windows Server 2022 image](#)

Windows Server 2019

Unless you need to run Windows Server 2012 R2 or Windows Server 2016 for some reason, we recommend using the latest version of Windows Server 2019.

Lightsail running Windows Server is a fast and dependable environment for deploying applications using the Microsoft Web Platform. Lightsail enables you to run any compatible Windows-based solution on AWS' high-performance, reliable, cost-effective, cloud computing platform. Common Windows use cases include Enterprise Windows-based application hosting, website and web-service hosting, data processing, distributed testing, ASP.NET application hosting, and any other application requiring Windows software.

[Learn more about the Windows Server 2019 image](#)

Windows Server 2016

Lightsail running Windows Server is a fast and dependable environment for deploying applications using the Microsoft Web Platform. Lightsail enables you to run any compatible Windows-based solution on AWS' high-performance, reliable, cost-effective, cloud computing platform. Common Windows use cases include Enterprise Windows-based application hosting, website and web-service hosting, data processing, distributed testing, ASP.NET application hosting, and any other application requiring Windows software.

[Learn more about the Windows Server 2016 image](#)

SQL Server Express 2022

SQL Server Express is a relational database management system that is free to download, distribute, and use. It comprises a database specifically targeted for embedded and smaller-scale applications. This Lightsail image runs on a base OS of Windows Server 2022.

[Learn more about the SQL Server Express 2022 image](#)

SQL Server Express 2019

SQL Server Express is a relational database management system that is free to download, distribute, and use. It comprises a database specifically targeted for embedded and smaller-scale applications. This Lightsail image runs on a base OS of Windows Server 2022.

[Learn more about the SQL Server Express 2019 image](#)

SQL Server Express 2016

SQL Server Express is a relational database management system that is free to download, distribute, and use. It comprises a database specifically targeted for embedded and smaller-scale applications. This Lightsail image runs on a base OS of Windows Server 2016.

[Learn more about the SQL Server Express image](#)

Create a Windows Server-based instance

You can create a Windows Server-based instance using the Lightsail console or by using the AWS Command Line Interface (AWS CLI).

To create an instance using the console

1. Sign in to Lightsail, and then go to the home page.

2. Choose **Create instance**.
3. Select an AWS Region where you want to create your Windows Server-based Lightsail instance.

For example, Ohio (us-east-2).

4. Select the **Microsoft Windows** platform.
5. To choose the Windows Server 2022, Windows Server 2019, Windows Server 2016 blueprint, choose **OS Only**.

To choose the SQL Server Express blueprint, choose **Apps + OS**.

6. Choose your instance plan.

Choose whether your instance uses dual-stack (IPv4 and IPv6), or IPv6-only networking. Some Lightsail blueprints don't support IPv6-only networking at this time. To see which blueprints support IPv6-only networking see [Choose an Amazon Lightsail instance image](#).

A plan also includes a low, predictable cost and a machine configuration (RAM, SSD, vCPU), as well as data transfer.

 **Note**

Some instance plans are not available for some blueprints. For example, you can't use the two smallest plans with the SQL Server Express blueprint. At a minimum, you must use the plan that has 2 GB RAM and 50 GB SSD, or choose one of the larger plans.

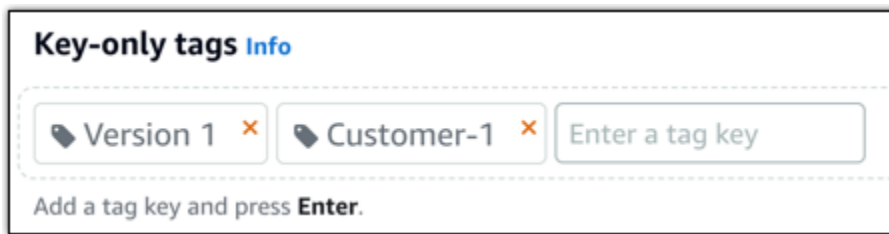
7. Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

8. Choose one of the following options to add tags to your instance:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



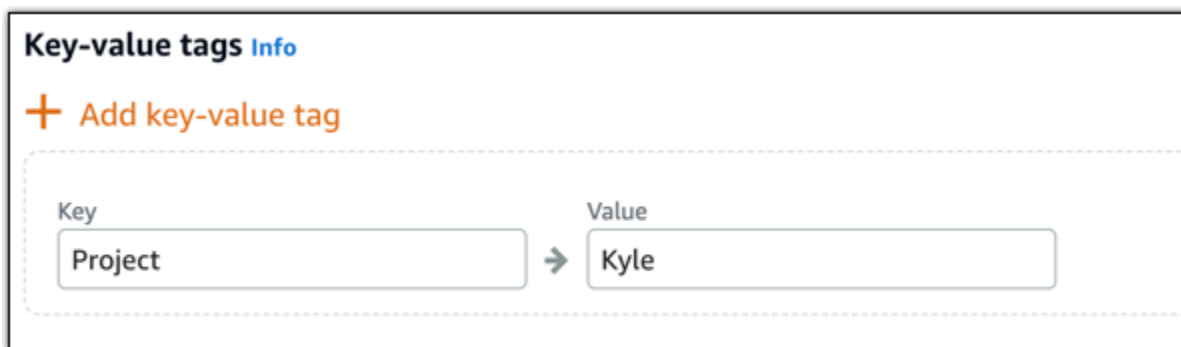
Key-only tags Info

Version 1 × Customer-1 × Enter a tag key

Add a tag key and press **Enter**.

- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key: Project → Value: Kyle

Note

For more information about key-only and key-value tags, see [Tags](#).

9. Choose **Create instance**.

To create an instance using the AWS CLI

1. If you haven't done so already, install and configure the AWS CLI.

For more information, see [Configure the AWS Command Line Interface to work with Amazon Lightsail](#).

2. Open a command prompt or a terminal window.
3. If you haven't done so already, configure the AWS CLI using `aws configure` and select the AWS Region where you want to create your Lightsail resources.

4. Type the following AWS CLI command to create a \$40 USD per month Windows Server 2016 instance running in the Ohio region:

```
aws lightsail create-instances --instance-names InstanceName --availability-zone us-east-2a --blueprint-id windows_server_2016_2017_09_13 --bundle-id medium_win_1_0
```

In the command, replace *InstanceName* with the name of your new instance.

If successful, you'll see the following output from the AWS CLI.

```
{
  "operations": [
    {
      "status": "Started",
      "resourceType": "Instance",
      "isTerminal": false,
      "statusChangedAt": 1508086226.4,
      "location": {
        "availabilityZone": "us-east-2a",
        "regionName": "us-east-2"
      },
      "operationType": "CreateInstance",
      "resourceName": "my-windows-instance",
      "id": "344acdc8-f9c4-4eda-8232-12345EXAMPLE",
      "createdAt": 1508086225.467
    }
  ]
}
```

Note

To get a list of available blueprints, use the [get-blueprints](#) command. To get a list of available bundles, use the [get-bundles](#) command. Learn more about getting the password for your instance using the [get-instance-access-details](#) command.

Connect to your instance

Once you create your Windows Server-based Lightsail instance, you can connect to it using either the browser-based RDP client or the remote desktop client of your choice.

Note

After you create your instance, you may need to wait up to 15 minutes before you can connect to it.

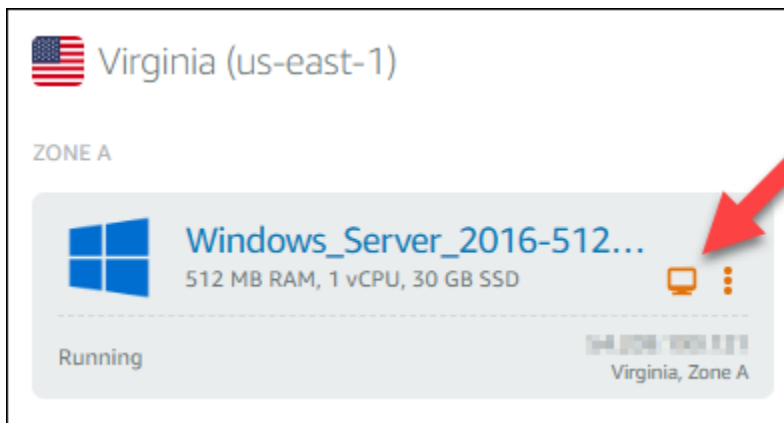
To connect using the Lightsail browser-based RDP client

1.

Note

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#)

On the home page, choose the **Connect using RDP** icon next to your instance.

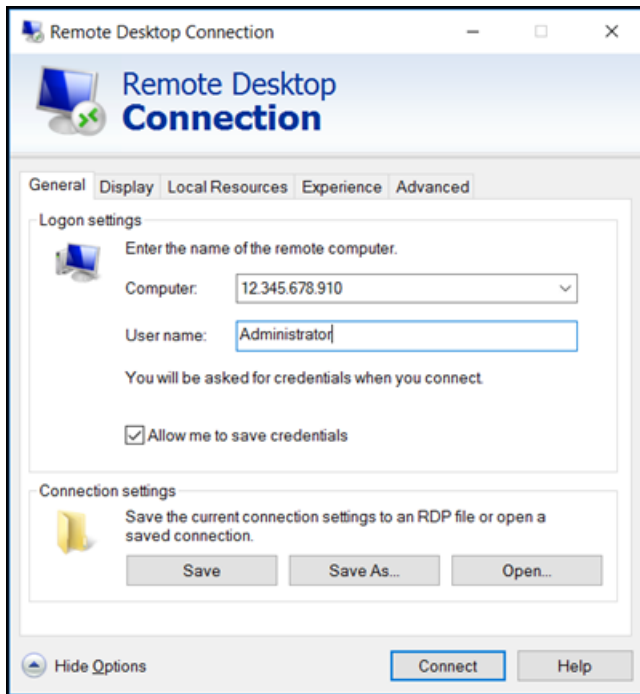


2. Alternately, you can connect to your instance from the shortcut menu or the instance management page.

To connect using your own RDP client

1. To get your IP address, go to the Lightsail home page.
2. Copy the IP address to the clipboard.

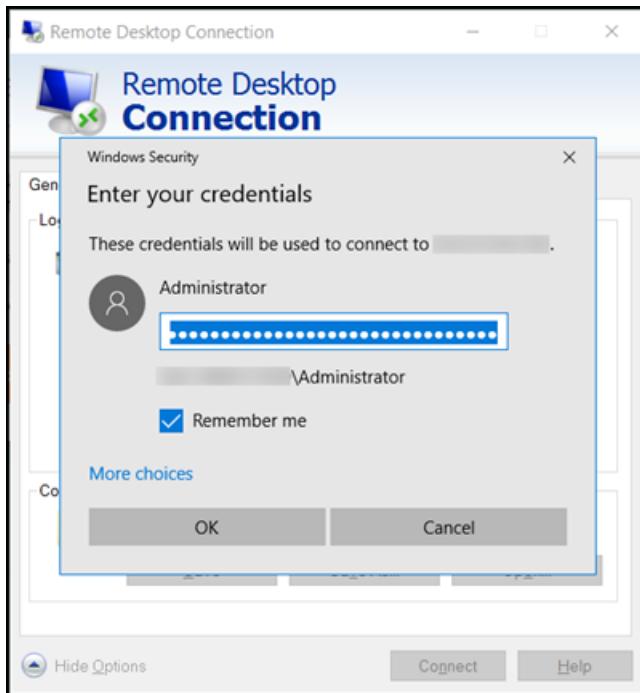
3. Open an RDP client such as **Remote Desktop Connection** in Windows.
4. Paste the IP address into the **Computer** field.
5. Choose **Show Options**, and then type Administrator for your **User name**.



6. Choose **Connect**.
7. To get your password, go to the instance management page in Lightsail.

You can get to the instance management page by choosing the name of your instance (or choosing **Manage** from the shortcut menu) on the Lightsail home page.

8. Choose **Show default password**.
9. Copy the default password to the clipboard.
10. Paste your password into **Remote Desktop Connection**, and then choose **Remember me** to prevent this dialog box from appearing in the future.



11. Choose **OK**.
12. Choose **Don't ask me again for connections to this computer**, and then choose **Yes**.

Instances (virtual private servers) in Amazon Lightsail

Your Lightsail instance is a virtual private server (also called a *virtual machine*). When you create your instance, you choose an image that has an operating system (OS) on it. You can also choose an instance image that has an application or development stack on it, including the base OS.

For a complete list of operating systems, applications, and development frameworks, see [Choose a Lightsail instance image](#).

See the following topics for more information about instances:

Topics

- [Create a Lightsail instance](#)
- [Delete a Lightsail instance](#)
- [Choose an Amazon Lightsail instance image](#)
- [IPv6-only instance plans in Lightsail](#)
- [SSH key pairs in Lightsail](#)
- [Create a snapshot of your Linux or Unix Lightsail instance](#)
- [Manage your Lightsail instance](#)
- [Lightsail firewall rules reference](#)
- [Instance Metadata Service \(IMDS\) and user data in Lightsail](#)

Create a Lightsail instance

You can create a Lightsail instance, also known as a virtual private server (VPS), running an application like WordPress or a development stack like LAMP in seconds. After your instance starts running, you can connect to it via SSH without leaving Lightsail. Here's how.

1. On the home page, choose **Create instance**.
2. Select a location for your instance (an AWS Region and Availability Zone).

Choose **Change AWS Region and Availability Zone** to create your instance in another location.

3. Optionally, you can change the Availability Zone.

Choose an Availability Zone from the dropdown list.

- Pick an application (**Apps + OS**) or an operating system (**OS Only**).

To learn more about Lightsail instance images, see [Choose an Amazon Lightsail instance image](#).

- Choose your instance plan.

Choose whether your instance uses dual-stack (IPv4 and IPv6), or IPv6-only networking. Some Lightsail blueprints don't support IPv6-only networking at this time. To see which blueprints support IPv6-only networking see [Choose an Amazon Lightsail instance image](#).

You can try the \$3.50 USD Lightsail plan free for one month (up to 750 hours). We will credit one free month to your account. Learn more on our [Lightsail pricing page](#).

Note

As part of the AWS Free Tier, you can get started with Amazon Lightsail for free on select instance bundles. For more information, see **AWS Free Tier** on the [Amazon Lightsail Pricing page](#).

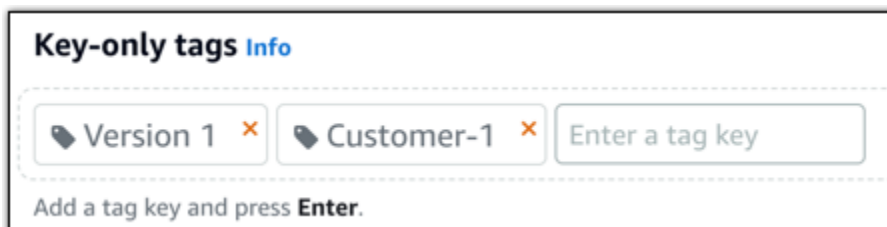
- Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

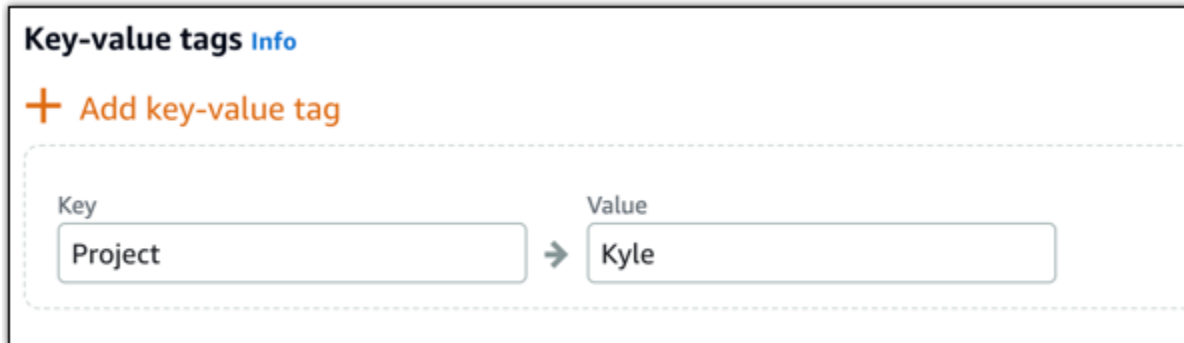
- Choose one of the following options to add tags to your instance:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



The screenshot shows a user interface for adding key-value tags. At the top, it says "Key-value tags Info". Below that is a plus sign icon followed by the text "Add key-value tag". Underneath is a dashed-line box containing two input fields. The first field is labeled "Key" and contains the text "Project". To its right is a right-pointing arrow, followed by a second input field labeled "Value" which contains the text "Kyle".

Note

For more information about key-only and key-value tags, see [Tags](#).

8. Choose **Create instance**.

For advanced creation options, see [Use a launch script to configure your Amazon Lightsail instance when it starts up](#) or [Set up SSH for your Linux/Unix-based instances](#).

Within minutes, your Lightsail instance is ready and you can connect to it via SSH, without leaving Lightsail!

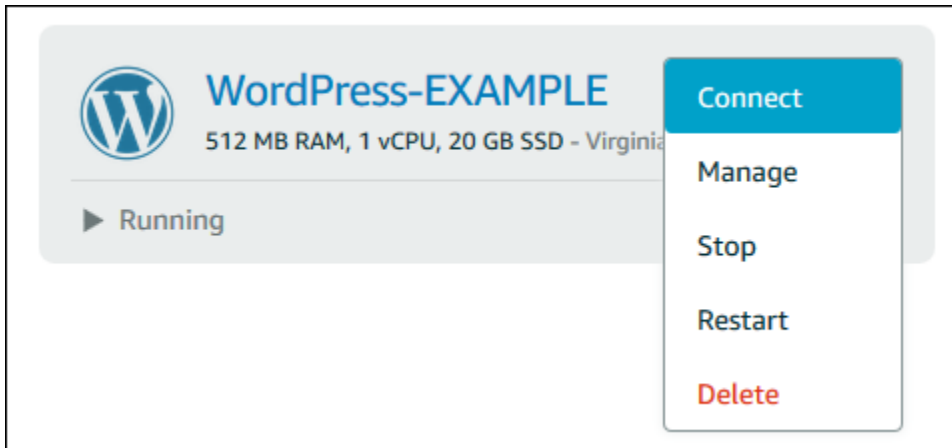
How to connect to your instance

- 1.

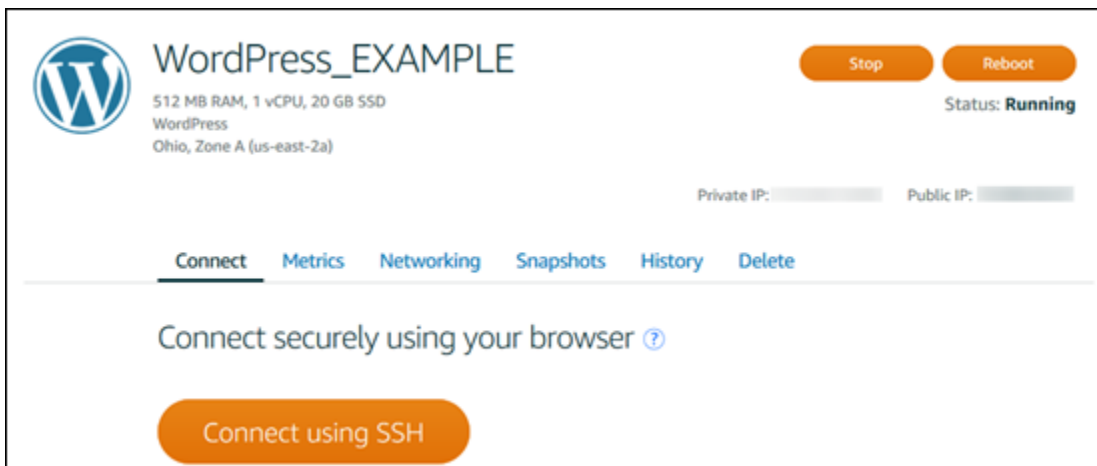
Note

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#)

From the Lightsail home page, choose the menu on the right of your instance's name, and then choose **Connect**.



Alternately, you can open your instance management page and choose the **Connect** tab.



Note

If you want to connect to your instance using an SSH client such as PuTTY, you can follow this procedure: [Set up PuTTY to connect to your Lightsail instance](#).

2. You can now type commands into the terminal and manage your Lightsail instance without setting up an SSH client.

```

WordPress-512MB-Ireland-1 - Terminal | Lightsail - Google Chrome
Secure | https://lightsail.aws.amazon.com/ls/terminal/eu-west-1/WordPress-512MB-Ireland-1?protocol...
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)
*** System restart required ***

      _ _ _
     / _ \| | |
    / ___ \| |_|
   /_/   \|_||_|

*** Welcome to the Bitnami WordPress 4.9.6-0 ***
*** Documentation: https://docs.bitnami.com/aws/apps/wordpress/ ***
*** https://docs.bitnami.com/aws/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***
Last login: Fri Jun 15 19:57:10 2018 from [REDACTED]
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bitnami@ip-192-168-1-23:~$
  
```

Next steps

Now that you can connect to your instance, what you do next depends on how you plan to use it. For example:

- [the section called “WordPress”](#) if you're creating a blog.
- [Create a static IP address](#) for your instance to keep the same IP address each time you restart your Lightsail instance.
- [Create a snapshot of your instance](#) as a backup.

Delete a Lightsail instance

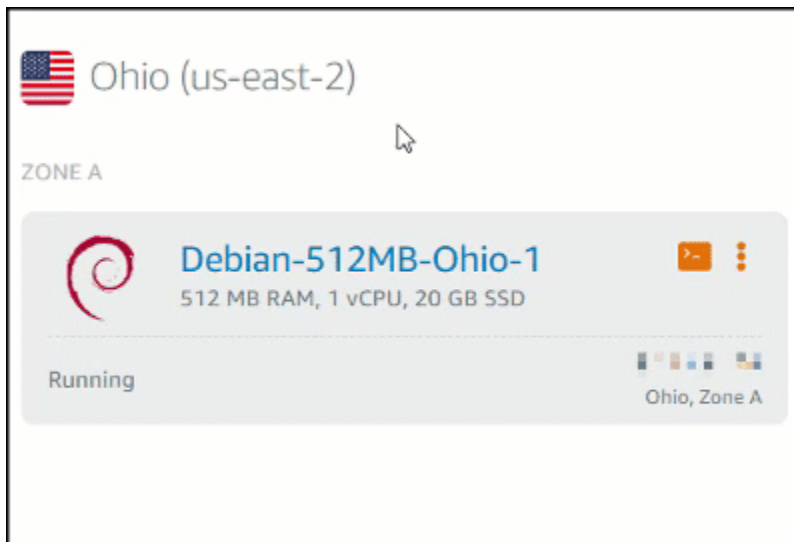
If you no longer need an instance, you can delete it using the Amazon Lightsail console or the AWS Command Line Interface (AWS CLI). You stop incurring charges for the instance as soon as it's deleted. However, resources attached to the deleted instance, such as static IPs and snapshots, continue to incur charges until you delete them.

Note

Deleted instances cannot be recovered. Create a snapshot of an instance before deleting if you might need the data in the instance at a later time. For more information, see [Create a snapshot of your Linux or Unix instance](#) or [Create a snapshot of your Windows Server instance in](#).

Delete an instance from the Lightsail console home page

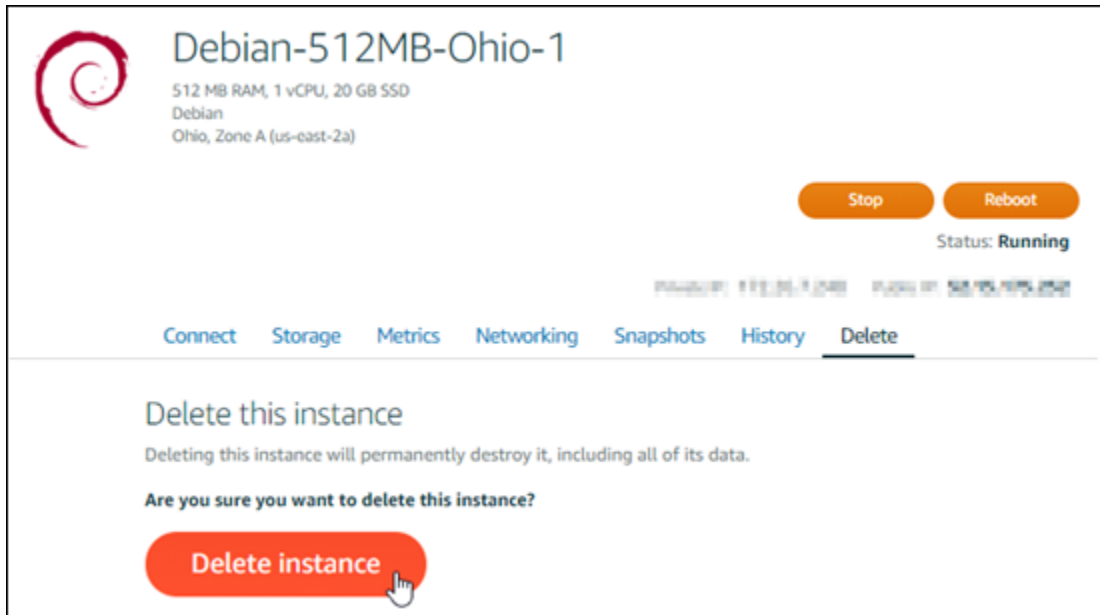
1. Sign in to the [Lightsail console](#).
2. For the instance you want to delete, choose the actions menu icon (:), then choose **Delete**.



3. Choose **Yes** to confirm the deletion.

Delete an instance from the Lightsail console instance management page

1. In the Lightsail console on the home page, choose the instance you want to delete.
2. Choose the **Delete** tab, then choose **Delete instance**.



3. Choose **Yes** to confirm the deletion.

Delete an instance using the AWS CLI

1. Complete the following prerequisites if you haven't already:
 - a. Install the AWS CLI. For more information, see [Install the AWS CLI](#).
 - b. Configure the AWS CLI. For more information, see [Configuring the AWS CLI](#).
2. Open a Terminal or Command Prompt window, then type the following command to get the name of the instance you want to delete:

```
aws lightsail get-instances
```

You should see results similar to the following:

```
C:\>aws lightsail get-instance --instance-name Ubuntu-512MB-Ohio-1
{
  "instance": {
    "username": "ubuntu",
    "isStaticIp": false,
    "networking": {
      "monthlyTransfer": {
        "gbPerMonthAllocated": 1024
      },
      "ports": [
        {
          "protocol": "tcp",
          "accessType": "public",
          "commonName": "",
          "accessFrom": "Anywhere (0.0.0.0/0)",
          "fromPort": 80,
          "accessDirection": "inbound",
          "toPort": 80
        },
        {
          "protocol": "tcp",
          "accessType": "public",
          "commonName": "",
          "accessFrom": "Anywhere (0.0.0.0/0)",
          "fromPort": 22,
          "accessDirection": "inbound",
          "toPort": 22
        }
      ]
    },
    "name": "Ubuntu-512MB-Ohio-1",
    "resourceType": "Instance",
    "supportCode": "LIGHTSAIL-INSTANCES-512MB-OHIO-1",
    "blueprintName": "Ubuntu",
    "hardware": {
      "cpuCount": 1,

```

3. Select and copy the name of the instance you want to delete so you can use it in the next step.

Note

If the instance you want to delete does not appear, confirm that your AWS CLI is configured for the AWS Region where the instance is located. For more information, see [Configuring the AWS CLI](#).

4. Type the following command to delete the instance.

```
aws lightsail delete-instance --instance-name InstanceName
```

In the command, replace *InstanceName* with the name of the instance.

If the deletion is successful, you should see a confirmation similar to the following:

```
C:\>aws lightsail delete-instance --instance-name Ubuntu-512MB-Ohio-1
{
  "operations": [
    {
      "status": "Succeeded",
      "resourceType": "Instance",
      "isTerminal": true,
      "statusChangedAt": 1527202978.962,
      "location": {
        "availabilityZone": "us-east-2a",
        "regionName": "us-east-2"
      },
      "operationType": "DeleteInstance",
      "resourceName": "Ubuntu-512MB-Ohio-1",
      "id": "aws-lightsail-1527202978-962-000000000000",
      "createdAt": 1527202978.962
    }
  ]
}
```

Note

If the deletion isn't successful, you should see an error message. Confirm that you copied and pasted the exact name of the instance and try again.

Next steps

After you delete an instance, a static IP, snapshots, block storage disks, and load balancer associated to an instance remain in Lightsail, and incur additional charges. For more information about how to delete those resources, see the following articles:

- [Delete a static IP](#)
- [Delete a snapshot](#)
- [Detach and delete a block storage disk](#)
- [Delete a load balancer](#)

Choose an Amazon Lightsail instance image

Lightsail provides several options for you to create your virtual private server. This topic helps you decide which operating system (OS), application, or development stack is right for your project. We organized the applications by functional area (such as CMS and ecommerce).

Compare platforms

Lightsail has two platforms to choose from: Linux/Unix-based or Windows-based platforms. If you already have an application in mind, you probably already have an OS platform picked out. You can choose one of the following options to get started:

- [Get started with Linux/Unix-based instances](#)
- [Get started with Windows-based instances](#)

Compare operating systems

Lightsail has several operating systems to choose from.

Windows Server 2022

Lightsail running Windows Server is a fast and dependable environment for deploying applications using the Microsoft Web Platform. With Lightsail, you can run any compatible Windows-based solution on the high-performance, reliable, cost-effective AWS Cloud computing platform. Common Windows use cases include Enterprise Windows-based application hosting, website and web service hosting, data processing, distributed testing, ASP.NET application hosting, and any other application requiring Windows software. For end of support information, see the [Microsoft website](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about the Windows Server 2022 image](#)

Windows Server 2019

Lightsail running Windows Server is a fast and dependable environment for deploying applications using the Microsoft Web Platform. Lightsail enables you to run any compatible Windows-based solution on the high-performance, reliable, cost-effective AWS cloud computing platform. Common Windows use cases include Enterprise Windows-based application hosting, website and web service hosting, data processing, distributed testing, ASP.NET application hosting, and any other application requiring Windows software. For end of support information, see the [Microsoft website](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about Windows Server 2019 image](#)

Windows Server 2016

Lightsail running Windows Server is a fast and dependable environment for deploying applications using the Microsoft Web Platform. Lightsail enables you to run any compatible Windows-based solution on the high-performance, reliable, cost-effective AWS cloud computing platform. Common Windows use cases include Enterprise Windows-based application hosting, website and web service hosting, data processing, distributed testing, ASP.NET application hosting, and any other application requiring Windows software. For end of support information, see the [Microsoft website](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about the Windows Server 2016 image](#)

Amazon Linux 2023

Amazon Linux 2023 (AL2023) is the next generation of Amazon Linux, ideal for general purpose workloads on AWS. AL2023 will be supported for five years after it is generally available. AL2023 locks to a specific version of the Amazon Linux package repository, giving you control over how and when you absorb updates. AL2023 also provides the ability to get frequent updates and comes with features to help you meet your compliance needs.

Lightsail instances launched from AL2023 will have Instance Metadata Service Version 2 (IMDSv2) enforced by default. For more information, see [How Instance Metadata Service Version 2 works](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about Amazon Linux 2023](#).

Amazon Linux 2

Amazon Linux 2 is the previous generation of Amazon Linux, a Linux server operating system from AWS. It provides a secure, stable, and high performance execution environment to develop and run cloud and enterprise applications. With Amazon Linux 2, you get an application environment that offers long term support with access to the latest innovations in Linux. Amazon Linux 2 is provided at no additional charge. For end of support information, see [Amazon Linux 2 FAQs](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about Amazon Linux 2.](#)

AlmaLinux OS 9

AlmaLinux OS 9 is an open source, community owned and governed, forever-free enterprise Linux distribution, focused on long-term stability, providing a robust production-grade platform. AlmaLinux is compatible with RHEL® and pre-Stream CentOS. For end of support information, see the [AlmaLinux OS Foundation](#) website.

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about AlmaLinux OS 9](#)

CentOS 7

Important

CentOS 7 will reach End of Lifetime (EOL) on June 30, 2024. You will not be able to create new Lightsail instances with this blueprint on or after June 30, 2024. For more information, see the [CentOS website](#).

CentOS is a Linux distribution that provides a free, enterprise-class, community-supported computing platform functionally compatible with its upstream source, Red Hat Enterprise Linux. For end of support information, see the [Red Hat website](#).

[Learn more about CentOS 7.](#)

CentOS Stream 9

CentOS Stream 9 is the next major release of the CentOS Stream distribution. CentOS Stream 9 is a continuously delivered distribution that tracks just ahead of Red Hat Enterprise Linux (RHEL) development, positioned as a midstream between Fedora Linux and RHEL. It's designed to be functionally compatible with RHEL and provides a stable, predictable, manageable and reproducible Linux environment. For end of support information, see the [CentOS website](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about CentOS Stream.](#)

Debian 10, 11 and 12

Important

Debian 10 will reach end of long term support on June 30, 2024. You will not be able to create new Lightsail instances with this blueprint on or after June 30, 2024.

Debian is a free operating system, developed by thousands of volunteers from all over the world who collaborate over the internet. The Debian project's key strengths are its volunteer base, its dedication to the Debian Social Contract and Free Software, and its commitment to provide the best operating system possible. This new release is another important step in that direction. For end of support information, see the [Debian website](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about Debian.](#)

FreeBSD 13

FreeBSD is an operating system used to power servers, desktops, and embedded systems. Derived from BSD, the version of UNIX developed at the University of California, Berkeley, FreeBSD has been continually developed by a large community for more than 30 years. FreeBSD's networking, security, storage, and monitoring features, including the pf firewall, the Capsicum and CloudABI capability frameworks, the ZFS file system, and the DTrace dynamic tracing framework, make FreeBSD the platform of choice for many of the busiest websites and most pervasive embedded networking and storage systems. For end of support information, see the [FreeBSD website](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about FreeBSD.](#)

openSUSE 15

The openSUSE distribution is a stable, easy to use and complete multipurpose Linux distribution. It is aimed towards users and developers working on the desktop or server. It is great for beginners, experienced users and ultra geeks alike, in short, it is perfect for everybody! For end of support information, see the [openSUSE website](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about openSUSE.](#)

Ubuntu 18, and 20, and 22

Important

Ubuntu 18.04 reached End of Standard Support on May 31, 2023. You will not be able to create new Lightsail instances with this blueprint on or after May 31, 2024. For more information, see the [Ubuntu website](#).

Ubuntu Server is a Debian-based Linux operating system used for virtual servers. A default installation of Ubuntu contains a wide range of software that includes LibreOffice, Firefox, Thunderbird, and Transmission. You can install many additional software packages, such as Evolution, GIMP, Pidgin, and Synaptic by using the APT-based package management tool (apt-get). For end of support information, see the [Ubuntu website](#).

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about Ubuntu.](#)

Compare Database applications

The following database applications are available in Lightsail:

SQL Server 2022 Express

SQL Server Express is a relational database management system that is free to download, distribute, and use. It comprises a database specifically targeted for embedded and smaller-scale applications. This Lightsail image runs on a base OS of Windows Server 2022.

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about the SQL Server 2022 Express image](#)

SQL Server 2019 Express

SQL Server Express is a relational database management system that is free to download, distribute, and use. It comprises a database specifically targeted for embedded and smaller-scale applications. This Lightsail image runs on a base OS of Windows Server 2022.

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about the SQL Server 2019 Express image](#)

SQL Server 2016 Express

SQL Server Express is a relational database management system that is free to download, distribute, and use. It comprises a database specifically targeted for embedded and smaller-scale applications. This Lightsail image runs on a base OS of Windows Server 2016.

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about the SQL Server 2016 Express image](#)

Compare CMS applications

The following content management system (CMS) applications are available in Lightsail:

WordPress certified by Bitnami

Bitnami WordPress is a preconfigured, ready-to-use image for running WordPress on Lightsail. WordPress is a popular web publishing platform for building blogs and websites. You can customize it by using a wide selection of themes, extensions, plugins, and widgets.

WordPress features a full theme system, which enables you to change the look and feel of your site with a few clicks. You can also use existing free or commercial WordPress themes. WordPress is in full compliance with the standards of the W3C.

[Learn more about the Bitnami WordPress application.](#)

WordPress Multisite certified by Bitnami

WordPress Multisite enables administrators to host and manage multiple websites from the same WordPress instance. These websites can all have unique domain names and can be customized by their owners, while sharing assets such as themes and plugins that are made available by the server admin. Updates to all sites can be pushed at once, ensuring that they are always kept safe and secure.

WordPress Multisite is great for organizations such as universities, corporations, and agencies that need to enable many people to host their own websites while giving overall control to a central administrator.

[Learn more about the Bitnami WordPress Multisite application.](#)

cPanel & WebHost Manager (WHM)

cPanel & WHM is a suite of tools built for Linux OS that gives you the ability to automate web hosting tasks by using a simple graphical user interface. Its goal is to make managing servers easier for you and managing websites easier for your customers.

[Learn more about cPanel & WHM.](#)

PrestaShop packaged by Bitnami

PrestaShop is one of the most prolific ecommerce solutions in the world. It is free and open source software, with a community of over 1 million active members. It is designed to get your online store up and running quickly, with a preconfigured theme so that you can start selling almost immediately along with a Live Configurator for easily customizing the look of your site. PrestaShop features multi-store support, customizable URLs, multiple payment gateway options (including PayPal and Stripe), and marketplace integration with Amazon, eBay, Facebook and more.

[Learn more about PrestaShop.](#)

Ghost packaged by Bitnami

Ghost is a publishing platform that is suitable for everything from personal blogs to major news websites. Built on Node.js, its modern technology stack makes it versatile and flexible for developers seeking to integrate with other applications and tools, while maintaining ease of use for content creators.

[Learn more about the Bitnami Ghost application.](#)

Joomla! packaged by Bitnami

Bitnami Joomla! is a preconfigured, ready-to-use image for running Joomla! on Lightsail. Joomla! is a CMS that you can use to build a variety of websites or portals. These include personal, corporate, small business, nonprofit, and other organizational websites.

Joomla! also features a registration system that enables users to configure personal options. Authentication is an important part of user management, and Joomla! supports multiple protocols, including LDAP, OpenID, and others. Joomla! supports many different languages and offers guidance for using them for the website and the administration panel. Also, the **Banner Manager** makes it easy to set up and manage banners on your site. You can track metrics, including setting impression numbers, special URLs, and more.

[Learn more about the Bitnami Joomla! application.](#)

Drupal packaged by Bitnami

Bitnami Drupal is a preconfigured, ready-to-use image for running Drupal on Lightsail. Drupal is a content management platform that helps users easily publish, manage, and organize content. It's used for community web portals, discussion sites, corporate websites, and more. You can easily extend Drupal by plugging in modules. Drupal is built for high performance, is scalable to many servers, and has easy integration with REST, JSON, SOAP, and other formats.

There are thousands of add-on modules and designs available for Drupal free of charge. Drupal is also available in multiple languages.

[Learn more about the Bitnami Drupal application.](#)

Compare application stacks and servers

Lightsail has five application stacks and servers for a wide variety of development projects. Each image uses Linux/Unix (Ubuntu) as the base operating system.

LAMP stack (PHP 8) packaged by Bitnami

The Bitnami LAMP stack simplifies the development and deployment of PHP applications. It includes ready-to-run versions of Apache, MySQL, PHP, and phpMyAdmin, and also the other software required to run each of those components. Bitnami LAMP stack is completely integrated and configured, so you'll be ready to start developing your application as soon as you create your instance in Lightsail. Bitnami LAMP stack is regularly updated to ensure that you always have access to the latest stable releases for each bundled component.

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about the Bitnami LAMP stack.](#)

Django packaged by Bitnami

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Python is a dynamic object-oriented programming language that can be used for many kinds of software development. The Bitnami Django Stack greatly simplifies the deployment of Django and its runtime dependencies and includes ready-to-run versions of Python, Django, MySQL, and Apache.

[Learn more about the Bitnami Django stack.](#)

Node.js packaged by Bitnami

Bitnami Node.js is a preconfigured, ready-to-use image for running Node.js on Lightsail. Node.js is a platform built on Chrome's JavaScript runtime for easily creating fast, scalable network applications. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js is well suited for data-intensive, real-time applications.

[Learn more about the Bitnami Node.js stack.](#)

MEAN stack packaged by Bitnami

Bitnami MEAN stack provides a complete development environment for MongoDB and Node.js that you can deploy in one click. It includes the latest stable release of MongoDB, Express, Angular, Node.js, Git, PHP, and RockMongo.

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about the Bitnami MEAN stack.](#)

GitLab CE Packaged by Bitnami

Bitnami GitLab Community Edition (CE) is a preconfigured, ready-to-use image for running GitLab on Lightsail. GitLab is self-hosted Git management software that is fast, secure, and based on Ruby on Rails. GitLab CI (also included) is an open source Continuous Integration (CI) server closely integrated with Git and GitLab.

GitLab enables you to keep your code secure on your own server, manage repositories, users, and access permissions. It's self-contained, so you can duplicate or move the installation to different servers easily.

[Learn more about the Bitnami GitLab stack.](#)

Nginx (LEMP stack) packaged by Bitnami

Bitnami NGINX Stack provides a complete PHP, MySQL, and NGINX development environment that you can launch in one click. It also bundles phpMyAdmin, SQLite, ImageMagick, FastCGI, Memcache, GD, CURL, PEAR, PECL, and other components.

NGINX is an asynchronous server and its main advantage is scalability. The NGINX stack is also known as LEMP (Linux, NGINX, MySQL, and PHP).

[Learn more about the Bitnami Nginx \(LEMP\) stack.](#)

Plesk Hosting Stack on Ubuntu

Build, secure, and run websites and applications on Lightsail and AWS using the Hosting Stack powered by Plesk. This includes all your web-based server management and security tools, plus WordPress automation in a graphical user interface. It simplifies the work of web professionals and provides the scalability, security, and performance that your customers need.

[Set up and configure Plesk.](#)

[Learn more about the Plesk stack.](#)

Ecommerce applications

Lightsail currently has one ecommerce application image: Magento. This Magento image uses Linux/Unix (Ubuntu) as the base operating system.

Magento packaged by Bitnami

Bitnami Magento is a preconfigured, ready-to-use image for running Magento on Lightsail. You can build engaging, responsive, and secure sites using Magento. Magento is a feature-rich, flexible ecommerce solution that includes transaction options, multistore functionality, loyalty programs, product categorization, shopper filtering, promotion rules, and more.

You can use Magento to create a highly customized ecommerce site that reflects your brand. Magento integrates with your business operations, so you can manage your ecommerce site as your business needs.

[Learn more about the Bitnami Magento stack.](#)

Project management applications

Lightsail currently has one project management application image, Redmine. This image uses Linux/Unix (Ubuntu) as the base operating system.

Redmine packaged by Bitnami

Bitnami Redmine is a preconfigured, ready-to-use image for running Redmine on Lightsail. Redmine is a flexible project management web application. It includes support for multiple

projects, role-based access control, Gantt charts and calendars, management of news, documents, and files, per-project wikis and forums, SCM integration, and more.

This blueprint is compatible with a Lightsail IPv6-only instance plan.

[Learn more about the Bitnami Redmine stack.](#)

IPv6-only instance plans in Lightsail

Public, reachable IPv4 addresses are in short supply due to their widespread usage and constantly increasing global demand. The last available block of new IP version 4 (IPv4) addresses was allocated in 2011. Since that time, everyone has been reusing a finite set of available addresses. IP version 6 (IPv6) is the next generation IP address standard. IPv6 supplements—and will eventually replace—IPv4 in an attempt to remedy the depletion of IP addresses.

What are IPv6-only instance plans

Lightsail instance plans bundle an operating system (OS) and application of your choice. They also include support for both IPv4 and IPv6 (dual-stack), or IPv6-only networking. A dual-stack plan assigns a public IPv4 and a public IPv6 address to your instance. With this plan, you can enable or disable IPv6 as needed. With an IPv6-only instance plan, your instance receives a public IPv6 address and doesn't support public IPv4 traffic. To learn which Lightsail platforms and blueprints support IPv6-only plans, see [Choose an Amazon Lightsail instance image](#).

Create an IPv6-only instance if you don't require a public IPv4 address. Before you create an IPv6-only instance, ensure that you can communicate over IPv6. For more information, see IPv6 reachability in [Verify IPv6 reachability in Lightsail](#). To migrate an existing instance from dual-stack to IPv6-only, or from IPv6-only to dual-stack, see [Create a Lightsail instance from a snapshot](#).

IPv6 considerations

Review the following considerations before creating an IPv6-only instance:

- Ensure that your network infrastructure and Internet Service Provider (ISP) are both IPv6-compatible. For more information, see [Verify IPv6 reachability in Lightsail](#).
- Ensure that your application and users are able to communicate over IPv6. For more information, see [Verify IPv6 reachability in Lightsail](#).
- Your instance will communicate publicly over IPv6 only. It will also receive a private IPv4 address for communicating with other resources in your Lightsail account. IPv6-only instances do not

On Linux and Unix instances, the private key allows you to establish a secure SSH connection to your instance. On Windows instances, the private key decrypts the default administrator password that you use to establish a secure RDP connection to your instance.

Anyone who has access to your private key can connect to your instances, so it's important that you store your private key in a secure place.

Contents

- [Choosing a key pair option](#)
- [Connecting to your instances](#)
- [Manage keys stored on instances](#)

Choose a key pair option

You can choose one of the following key pair options when you create a Lightsail instance.

Windows instances always use the default key; therefore, you can't create a key pair or upload a key when creating Windows instances.

- **Default key pair** – Lightsail automatically creates a default key pair in each AWS Region where you create instances. When you use the default key pair with your instance, Lightsail stores the public key on your instance. You can download the private key of a default key pair at any time from the **Account** page on the Lightsail console. You can have up to one default key pair in each AWS Region.
- **Create key pair (Linux and Unix instances)** – You can use the Lightsail console to create a new custom key pair to use with your instance. When you create a custom key pair, you give it a unique name, and Lightsail stores the public key on your instance. You can download the private key of a custom key pair only when you first create it.
- **Upload key (Linux and Unix instances)** – To use an existing key pair of your own, you can upload your public key to Lightsail. When you upload a public key to use with your instance, you give it a unique name, and Lightsail stores it on your instance. You keep and store the private key of your key pair.

If you configure a single public key on multiple instances, you can use the same private key of the key pair to connect to those instances. For more information about managing key pairs, see [Managing key pairs in Amazon Lightsail](#).

Connect to your instances

You can connect to your Lightsail instances using one of the following options.

Lightsail browser-based SSH and RDP clients

In the Lightsail console, you can instantly connect to your Linux and Unix instances using a browser-based SSH client, and connect to your Windows instances using a browser-based RDP client. The Lightsail browser-based SSH and RDP clients only accept IPv4 traffic. Create a dual-stack instance, or use a third-party client to SSH or RDP into your instance over IPv6. You don't have to install an SSH client on your computer, configure key pairs, or specify administrator passwords when you connect to your instances using the browser-based clients. This is the fastest way to connect to your instances. For more information, see [Connecting to your Linux or Unix instance in Amazon Lightsail](#) and [Connecting to your Windows instance in Amazon Lightsail](#).

The browser-based clients use a different key pair than the one you configure when you create your instances, such as the default key, or a key you create or upload. Therefore, even if you delete or lose one of the keys you originally configured, you can continue to connect to your instances using the browser-based clients.

Third-party SSH and RDP clients

You can connect to your Linux and Unix instances using a third-party SSH client, and connect to your Windows instances using a third-party RDP client. When you use an SSH client, you must configure it to use the private key of the key pair that you configured on your instance. When you use an RDP client, you must specify the administrator password of your Windows instance.

If you use a Windows computer locally, you can use the following clients to connect to your Lightsail instances.

- **PuTTY** – Use PuTTY to connect to Linux or Unix instances using SSH. For more information, see [Set up PuTTY to connect to your instance](#).
- **Remote Desktop Connection** – Use the Remote Desktop Connection client to connect to Windows instances using RDP. For more information, see [Connect to your Windows instance using the Remote Desktop Connection client on a Windows computer](#).

If you use a Mac computer locally, use the following clients to connect to your Lightsail instances.

- **Native SSH client in Terminal** – Use the native SSH client in Terminal to connect to Linux and Unix instances. For more information, see [Connect to your Linux or Unix instance using SSH in Terminal](#).
- **Microsoft Remote Desktop** – Use the Microsoft Remote Desktop client for macOS to connect to Windows instances using RDP. For more information, see [Connect to your Windows instance using the Microsoft Remote Desktop client on a Mac](#).

Manage keys stored on instances

After your instance is up and running, you can add a new key to the instance, or replace the key that you originally assigned to it. For example, if a user in your organization requires access to the instance using a separate key, you can add that key to your instance. Another example might be when someone leaves your organization and they have a copy of the private key (.PEM) file. You can prevent them from connecting to your instance by replacing the key with a new one or removing it completely. For more information, see [Manage keys stored on an instance in Amazon Lightsail](#).

Topics

- [Connect to your Lightsail Linux or Unix instances](#)
- [Connect to your Lightsail Windows instance](#)

Connect to your Lightsail Linux or Unix instances

Amazon Lightsail provides you with a browser-based SSH client, which is the fastest way to connect to your Linux or Unix instance. You can also use your own SSH client to connect to your instance. For more information, see [Download and set up PuTTY](#).

Connect to your instance with SSH to perform administrative tasks on the server, such as installing software packages or configuring web applications. The browser-based SSH client requires no software installation, and is available almost immediately after you create an instance.

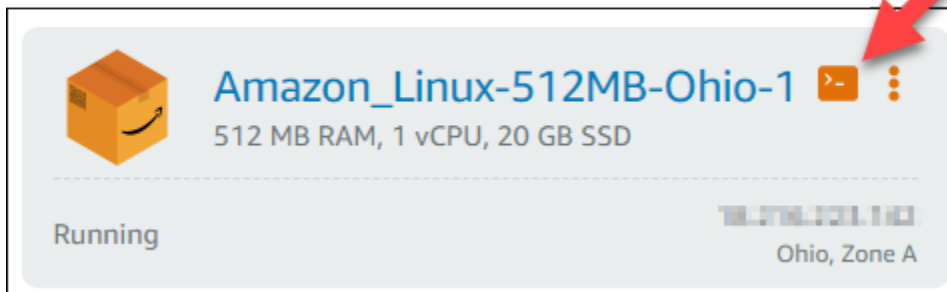
Note

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

To connect to a Windows Server instance in Lightsail, see [Connect to your Windows-based instance](#).

To connect to your Linux or Unix instance

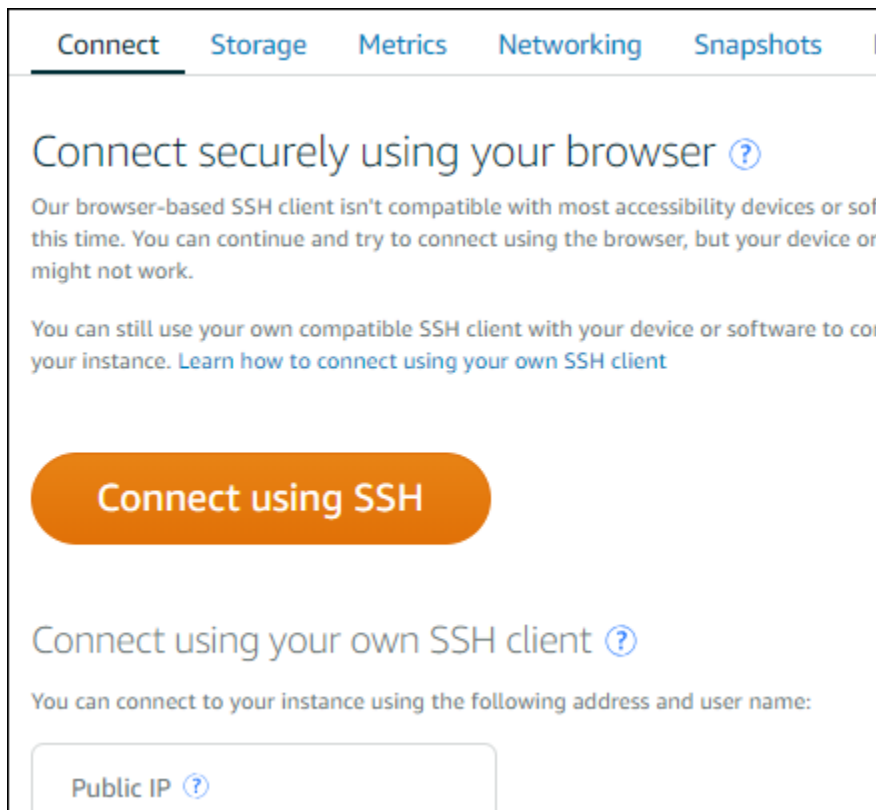
1. Sign in to the [Lightsail console](#).
2. Access the browser-based SSH client for the instance that you want to connect to by using any of the following:
 - Choose the quick connect icon, as shown in the following example.



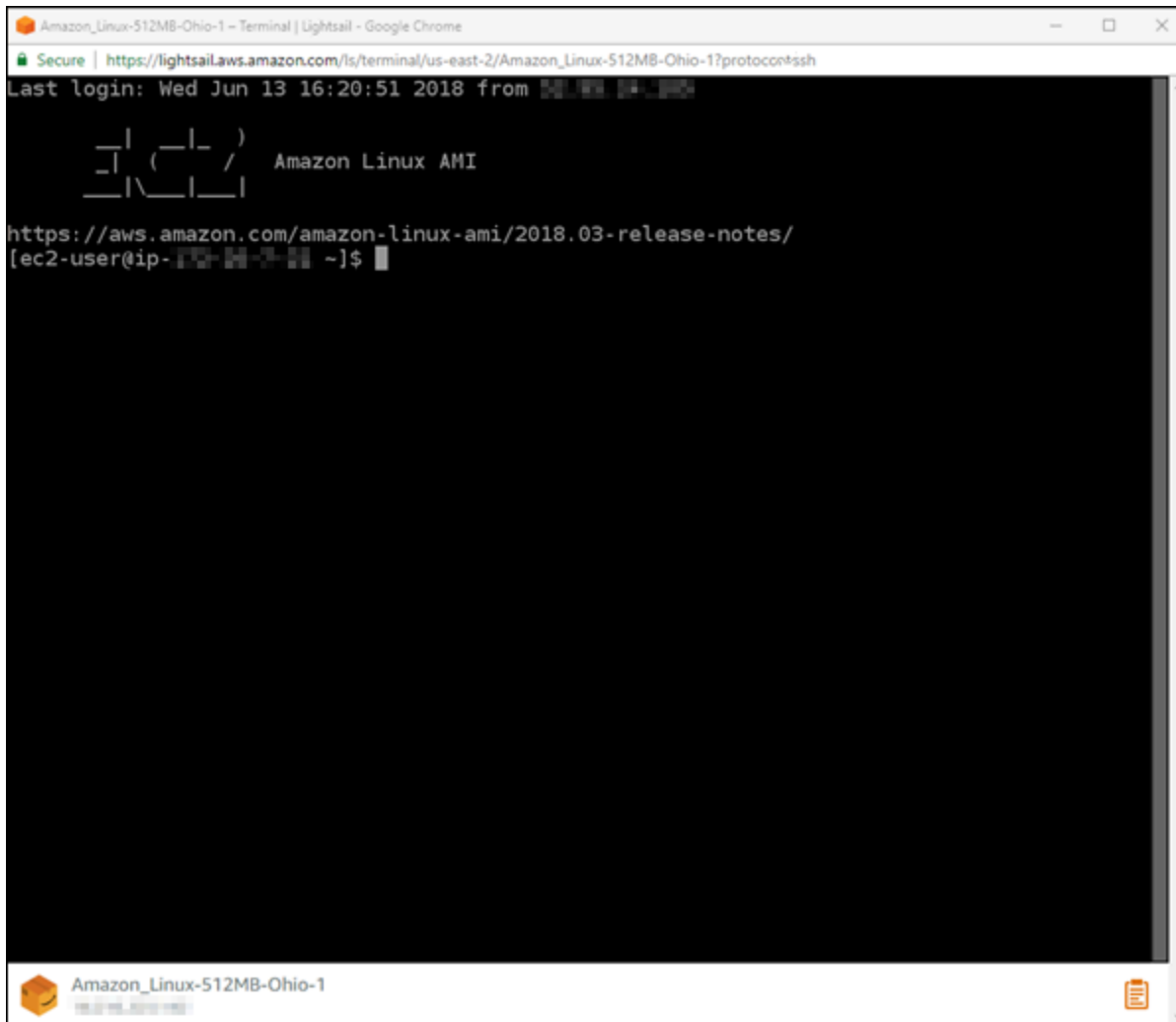
- Choose the actions menu icon (:), then choose **Connect**.



- Choose the name of the instance, and on the **Connect** tab, choose **Connect using SSH**.



You can start interacting with your instance when the browser-based SSH client opens, and a terminal screen is displayed as shown in the following example:

**Note**

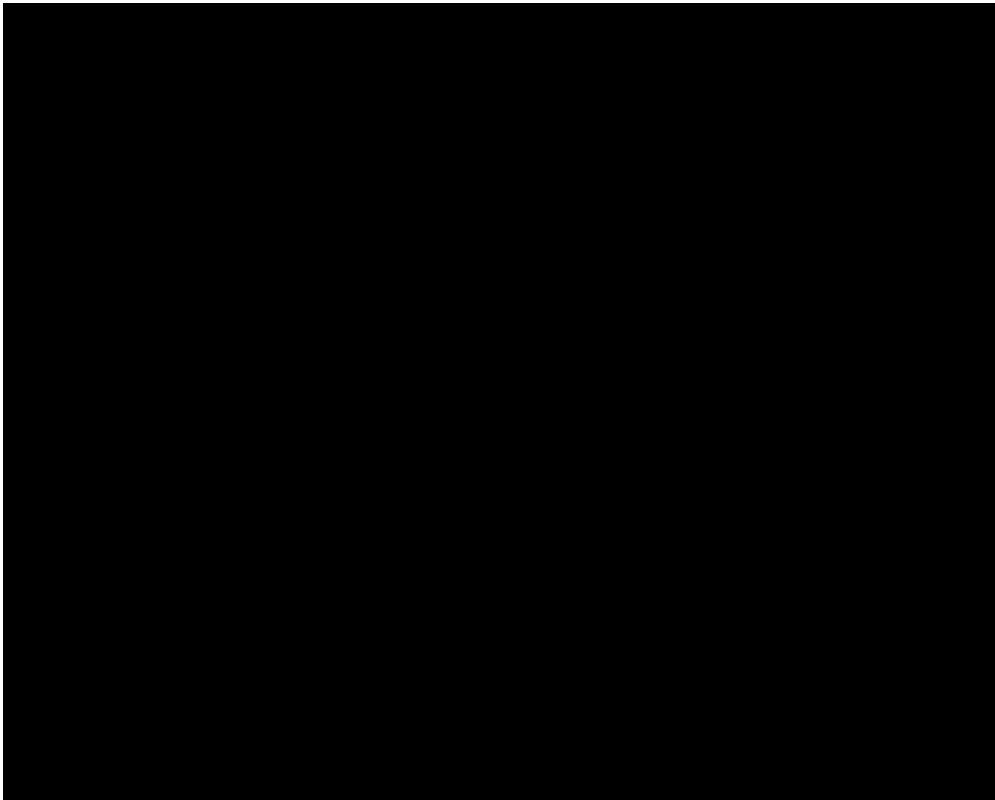
The **Connect** tab also provides the information required to connect using your own SSH client. For more information, see [Download and set up PuTTY](#)

Interact with your Linux or Unix instance using the browser-based SSH client

Type Linux or Unix commands directly into the terminal screen, paste text into the terminal screen, or copy text from the terminal screen of the browser-based SSH client. The following sections show you how to copy and paste text to and from the clipboard in SSH.

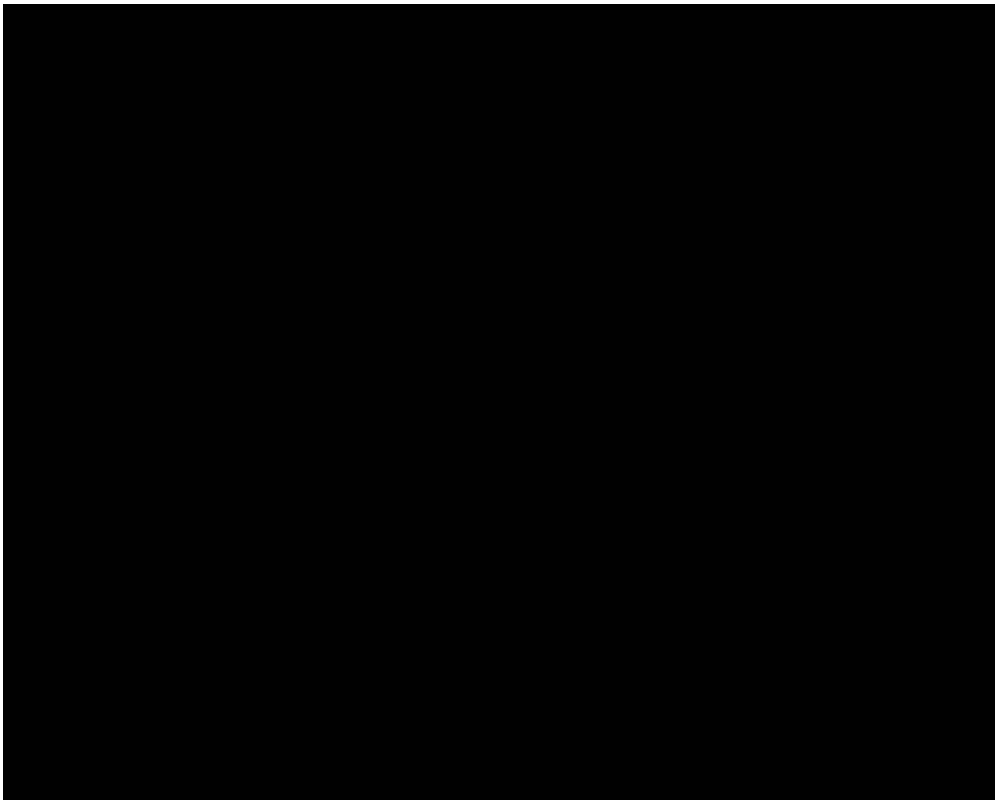
To paste text into the browser-based SSH client

1. Highlight text in your local desktop, then press **Ctrl+C** or **Cmd+C** to copy it to your local clipboard.
2. In the bottom right corner of the browser-based SSH client, choose the clipboard icon. The browser-based SSH client clipboard text box appears.
3. Click into the text box, then press **Ctrl+V** or **Cmd+V** to paste the contents from your local clipboard into the browser-based SSH client clipboard.
4. Right-click any area on the SSH terminal screen to paste the text from the browser-based SSH client clipboard to the terminal screen.



To copy text from the browser-based SSH client

1. Highlight text on the terminal screen.
2. In the bottom right corner of the browser-based SSH client, choose the clipboard icon. The browser-based SSH client clipboard text box appears.
3. Highlight the text that you want to copy, then press **Ctrl+C** or **Cmd+C** to copy the text to your local clipboard. You can now paste the copied text anywhere in your local desktop.



Set up SSH keys for Lightsail

Secure SHell (SSH) is a protocol for securely connecting to a virtual private server (or Lightsail *instance*). SSH works by creating a public key and a private key that match the remote server to an authorized user. Using that key pair, you can connect to your Lightsail instance using a browser-based SSH terminal.

For more information about SSH, see [Understanding SSH](#).

When you create your Lightsail instance, the default option is to let Lightsail manage your SSH keys for you. Lightsail provides a browser-based SSH client for securely connecting to your Linux-based instance. It's a fully functional terminal, where you can enter commands and make changes to your instance.

Windows-based instances use remote desktop (RDP) protocol instead of SSH. For more information about Windows-based instances in Lightsail, see [Get started with Windows-based instances in Lightsail](#).

⚠ Important

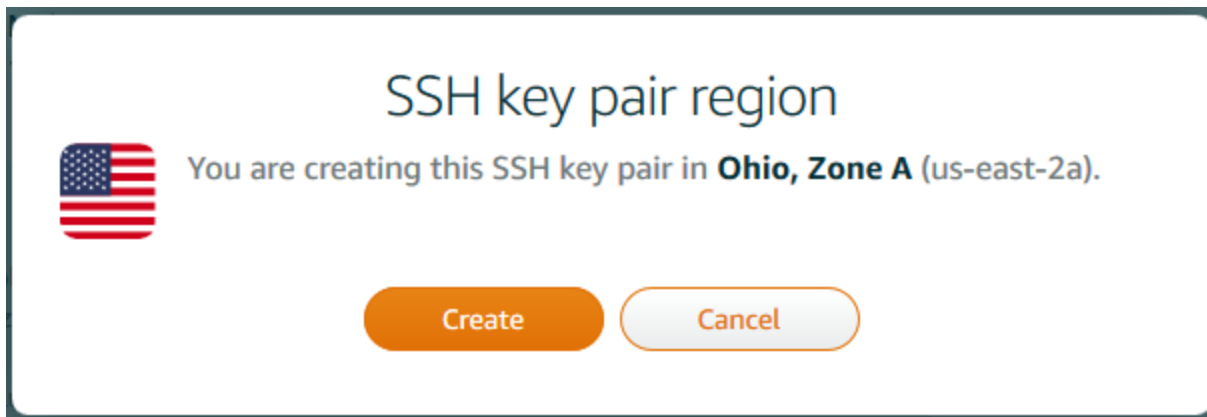
SSH key management is regional. When you create an instance in a new AWS Region, you will be given the option to use the default key pair for that region. You can also use a custom key in that region. Keep in mind that if you upload your own key, you'll have to do that for each region where you have a Lightsail instance.

If you use the default key, you can still download the private key for safekeeping. This can be done either at the time you create your instance or later. If you choose to download the key after you created your instance, you can do so under **SSH keys** on the **Account** page.

Create a new key

If you don't choose to use the default key, you can create a new key pair at the time you create your Lightsail instance.

1. If you haven't done it yet, choose **Create instance**.
2. On the **Create an instance** page, choose **change SSH key pair**.
3. Choose **Create new**.
4. Lightsail displays the region where we're creating the new key.



Choose **Create**.

5. Enter a name for your key pair.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.

- Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
6. Choose **Generate key pair**.

Important

Save your key somewhere you can easily find it. Also, it's a good idea to make sure permissions are set so that no one else can read it.

7. Continue creating your instance.

Upload an existing key

You can also choose to upload an existing key at the time you create your Lightsail instance.

1. If you haven't done it yet, choose **Create instance**.
2. On the **Create an instance** page, choose **change SSH key pair**.
3. Choose **Upload new**.
4. Lightsail displays the region where you're uploading the new key.

Choose **Upload**.

5. Choose **Browse** to find the key on your local machine.

Be sure to upload a public key (not a private key). For example, `github_rsa.pub`.

6. Choose **Upload key**.
7. Continue creating your instance.

Manage your keys

You can manage your keys on the **SSH keys** tab of the **Account** page. You will see each key pair in use in each region.

Profile **SSH keys** Advanced

SSH key pairs [?](#)

Choose your preferred key pair in each Region.
You can also create a new key pair or upload an existing key.

SSH key pairs can only be used in the Region where they are created or uploaded.

You may store up to 100 keys per Region.

Create New + Upload New

Virginia (us-east-1)

- Default** [?](#) Download
- custom.keypair
- Test_Keypair1

Oregon (us-west-2)

- Default** [?](#) Download
- github_rsa

Ohio (us-east-2)

- Default** [?](#) Download

On this page, you can change the key that should be used by default when you create new Lightsail instances. You can also create a new key, upload an existing key, or download a private key. You may want to use an SSH client like PuTTY to connect, which will require you to have the private half of the key. You can download the key on the **Account** page. [Learn more about setting up PuTTY to connect to a Lightsail instance.](#)

Connect to your Lightsail Linux/Unix-based instance using the SSH command

If your local machine uses a Linux or Unix operating system, including macOS, then you can connect to your Linux or Unix instance in Amazon Lightsail using the SSH client through a terminal window.

The method to connect to your instance described in this guide is one of many. For more information about the other methods, see [SSH key pairs](#).

The easiest way to connect to your Linux or Unix instance in Lightsail is by using the browser-based SSH client that is available in the Lightsail console. For more information, see [Connect to your Linux or Unix instance](#).

Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

Contents

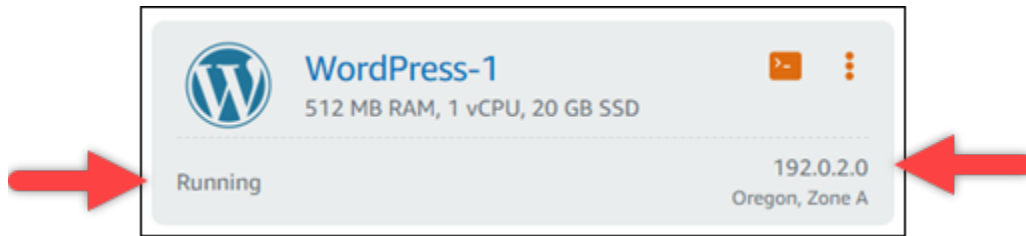
- [Step 1: Confirm your instance is running and get the public IP address](#)
- [Step 2: Confirm the SSH key pair being used by your instance](#)
- [Step 3: Change the permissions of your private key and connect to your instance using SSH](#)

Step 1: Confirm your instance is running and get the public IP address

In the following procedure, you sign in to the Lightsail console to confirm your instance is in a running state, and to get the public IP address of your instance. Your instance must be in a running state in order to establish an SSH connection, and you will need the public IP address of your instance to connect to it later in this guide.

1. Sign in to the [Lightsail console](#).
2. In the **Instances** tab of the Lightsail home page, locate the instance that you want to connect to.
3. Confirm that the instance is in a running state, and make note of the public IP address of your instance.

The state of your instance and its public IP address are listed next the name of your instance as shown in the following example.

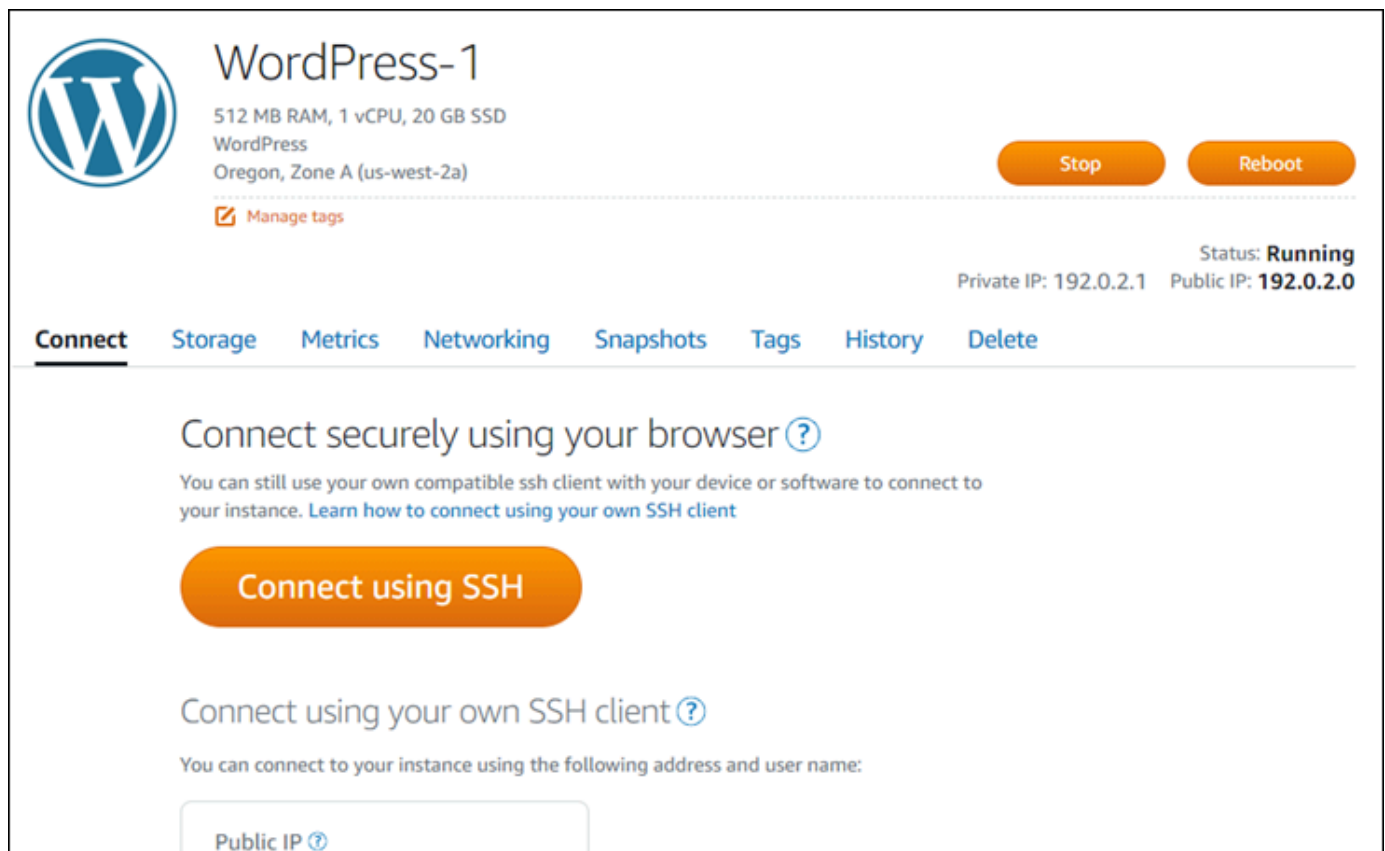


Step 2: Confirm the SSH key pair being used by your instance

In the following procedure you confirm the SSH key pair that is being used by your instance. You will need the private key of the key pair to authenticate to your instance and establish an SSH connection.

1. In the **Instances** tab of the Lightsail home page, choose the name of the instance that you want to connect to.

The **Instance management** page appears, with various tab options to manage your instance.



2. In the **Connect** tab, scroll down to see the key pair that is being used by your instance. There are two possibilities:

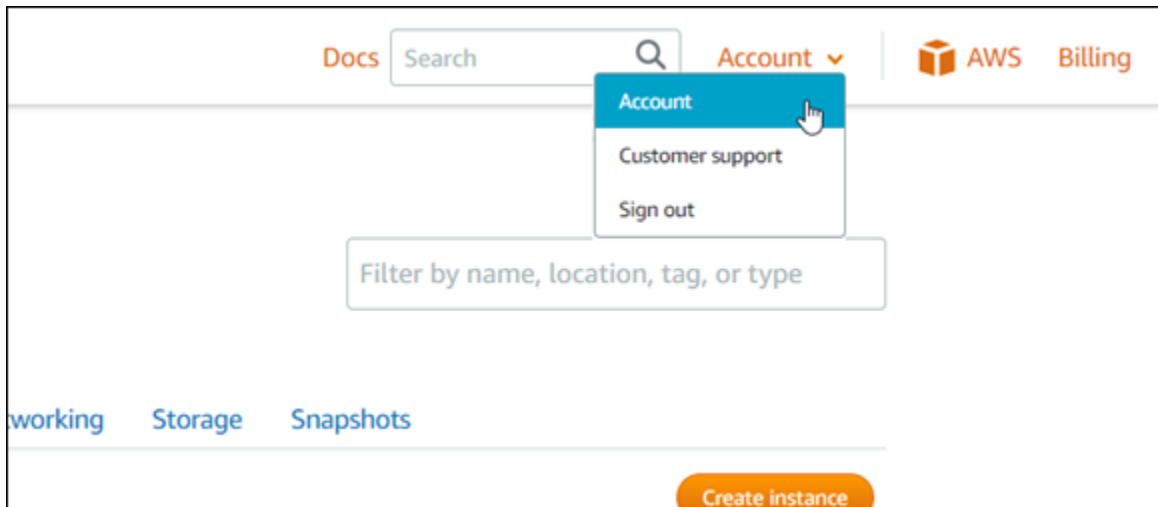
1. The following example shows an instance that uses the default key pair for the AWS Region in which you created your instance. If your instance is using the default key pair, then you can continue to step 3 of this procedure to download the private key of the key pair. Lightsail stores the private key only for the default key pair of each AWS Region.

You configured this instance to use **default (us-west-2)** key pair.
You can download your default private key from the [Account page](#).

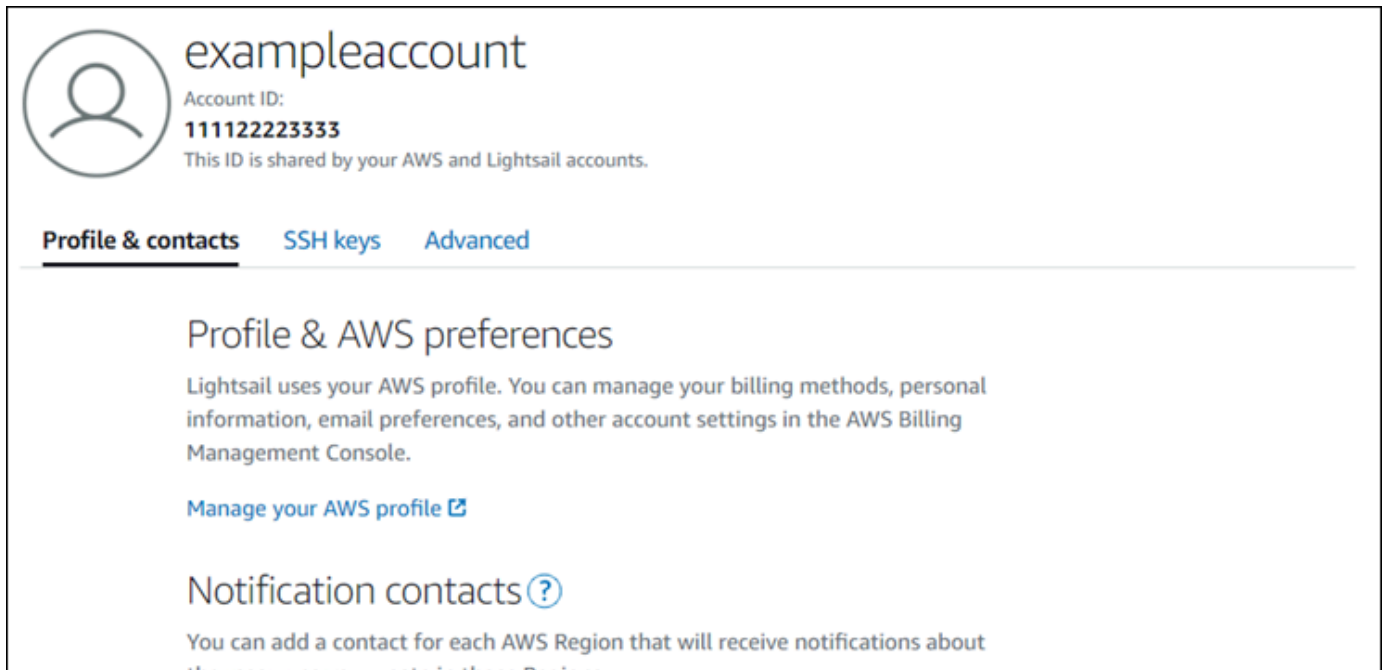
2. The following example shows an instance that uses a custom key pair that you either uploaded or created. If your instance is using a custom key pair, then you need to locate the private key of the custom key pair where you store your keys. If you lost the private key of the custom key pair, then you will not be able to establish an SSH connection to your instance using your own client. However, you can continue to use the browser-based SSH client available in the Lightsail console. Continue to the next [Step 3: Change the permissions of your private key and connect to your instance using SSH](#) section of this guide after you locate the private key of the custom key pair.

You configured this instance to use **MyKeyPair (us-west-2)** key pair.

3. Choose **Account** on the top navigation menu, then choose **Account**.

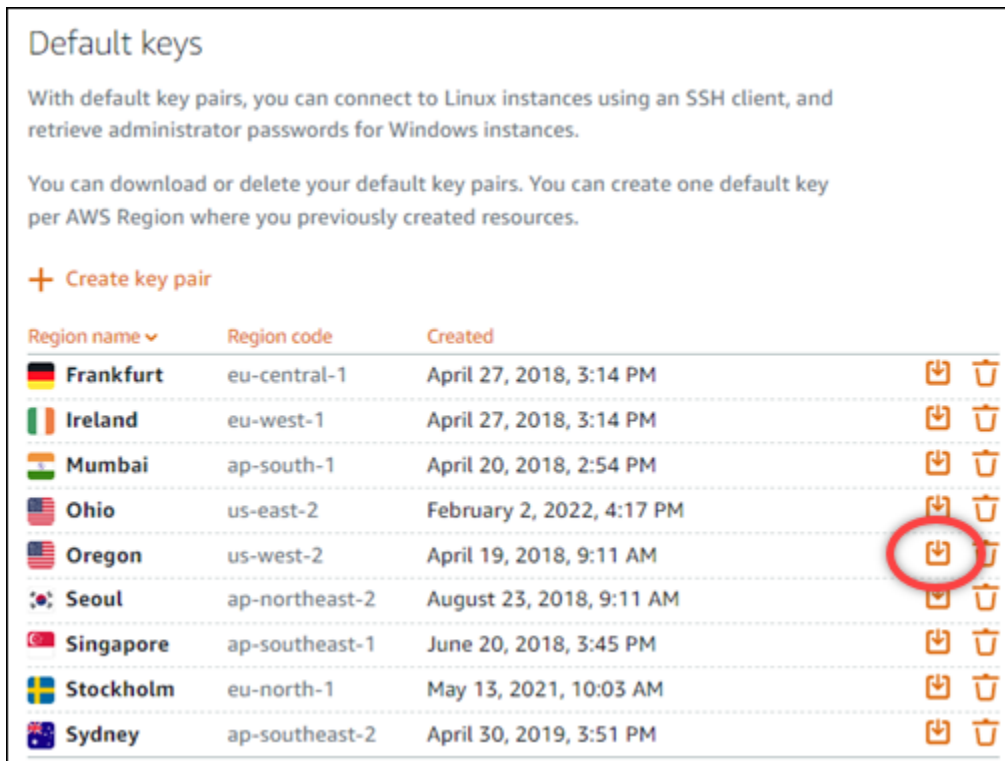


The **Account management** page appears, with various tab options to manage your account settings.



The screenshot shows the AWS account profile page for 'exampleaccount'. The account ID is 111122223333. Below the account information, there are three tabs: 'Profile & contacts', 'SSH keys', and 'Advanced'. The 'Profile & contacts' tab is selected. The main content area is titled 'Profile & AWS preferences' and contains a paragraph explaining that Lightsail uses the AWS profile for billing and account settings. A link 'Manage your AWS profile' is provided. Below this, there is a section for 'Notification contacts' with a help icon and a paragraph explaining that contacts receive notifications about resources in different regions.

4. Choose the **SSH keys** tab.
5. Scroll down, and choose the download icon next to the default key of the AWS Region of the instance that you want to connect to.



The screenshot shows the 'Default keys' page in the AWS console. It includes a heading 'Default keys', a paragraph explaining that default key pairs allow connecting to Linux instances and retrieving administrator passwords for Windows instances, and another paragraph stating that users can download or delete their default key pairs. A '+ Create key pair' button is visible. Below is a table of default key pairs with columns for 'Region name', 'Region code', and 'Created'. Each row has two icons: a download icon and a delete icon. The download icon for the 'Oregon' region is circled in red.

Region name	Region code	Created		
Frankfurt	eu-central-1	April 27, 2018, 3:14 PM	Download	Delete
Ireland	eu-west-1	April 27, 2018, 3:14 PM	Download	Delete
Mumbai	ap-south-1	April 20, 2018, 2:54 PM	Download	Delete
Ohio	us-east-2	February 2, 2022, 4:17 PM	Download	Delete
Oregon	us-west-2	April 19, 2018, 9:11 AM	Download	Delete
Seoul	ap-northeast-2	August 23, 2018, 9:11 AM	Download	Delete
Singapore	ap-southeast-1	June 20, 2018, 3:45 PM	Download	Delete
Stockholm	eu-north-1	May 13, 2021, 10:03 AM	Download	Delete
Sydney	ap-southeast-2	April 30, 2019, 3:51 PM	Download	Delete

The private key is downloaded to your local machine. You might want to move the downloaded key to a directory in which you store all of your SSH keys, such as a "Keys" folder

in your user's home directory. You will need to refer to the directory where the private key is saved in the next section of this guide. If the private key attempts to save as a format other than `.pem`, you should manually change the format to `.pem` before saving.

Note

Lightsail does not provide utilities for manipulating `.pem` files or other certificate formats. If you need to convert the format of your private key file, free and open-source tools such as [OpenSSL](#) are readily available.

Continue to the next [Step 3: Change the permissions of your private key and connect to your instance using SSH](#) section of this guide to use the private key you just downloaded and establish an SSH connection to your instance.

Step 3: Change the permissions of your private key and connect to your instance using SSH

In the following procedure you will change the permissions of your private key file to be readable and writable only by you. You then open a terminal window in your local machine, and run the SSH command to establish a connection with your instance in Lightsail.

1. Open a terminal window on your local machine.
2. Enter the following command to make the private key of the key pair readable and writable only by you. This is a security best practice required by some operating systems.

```
sudo chmod 400 /path/to/private-key.pem
```

In the command, replace `/path/to/private-key.pem` with the directory path to where you saved the private key of the key pair that is being used by your instance.

Example:

```
sudo chmod 400 /Users/user/Keys/LightsailDefaultKey-us-west-2.pem
```

3. Enter the following command to connect to your instance in Lightsail using SSH:

```
ssh -i /path/to/private-key.pem username@public-ip-address
```

In the command, replace:

- `/path/to/private-key.pem` with the directory path to where you saved the private key of the key pair that is being used by your instance.
- `username` with the username of your instance. You can specify one of the following user names depending on the blueprint that is used by your instance:
 - AlmaLinux OS 9, Amazon Linux 2, Amazon Linux 2023, CentOS Stream 9, FreeBSD, and openSUSE instances: `ec2-user`
 - CentOS 7 instances: `centos`
 - Debian instances: `admin`
 - Ubuntu instances: `ubuntu`
 - Bitnami instances: `bitnami`
 - Plesk instances: `ubuntu`
 - cPanel & WHM instances: `centos`
- Replace `public-ip-address` with the public IP address of your instance that you noted from the Lightsail console earlier in this guide.

Example with absolute path:

```
ssh -i /Users/user/Keys/LightsailDefaultKey-us-west-2.pem ec2-user@192.0.1.0
```

Example with relative path:

Notice the `./` prefixing the `.pem` file. Omitting `./` and simply writing `LightsailDefaultKey-us-west-2.pem` will not work.

```
ssh -i ./LightsailDefaultKey-us-west-2.pem ec2-user@192.0.1.0
```

You are successfully connected to your instance if you see the welcome message for your instance. The following example shows the welcome message for an Amazon Linux 2 instance; other instances blueprints have a similar welcome message. After you're connected, you can execute commands on your instance in Lightsail. To disconnect, enter `exit` and press Enter.

```
 _ | ( _ | - )
 _ | ( _ | - /
 _ | \ _ | _ |
                                     Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 13 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-0-10 ~]$
```

Connect to your Lightsail Linux/Unix-based instance using PuTTY

In addition to the browser-based SSH terminal in Lightsail, you can also connect to your Linux-based instance using an SSH client such as PuTTY. To learn how to set up PuTTY, see [Download and set up PuTTY to connect using SSH in Lightsail](#).

Note

To connect to a Windows-based instance using RDP, see [Connect to your Windows-based Lightsail instance](#).

You can use the default private key that Lightsail provides, a new private key from Lightsail, or another private key that you use with another service.

1. Start PuTTY (for example, from the **Start** menu, choose **All Programs, PuTTY, PuTTY**).
2. Choose **Load**, and then find your saved session.

If you don't have a saved session, see [Step 4: Finish configuring PuTTY with your private key and instance information](#).

3. Log in using one of the following default user names depending on your instance operating system:
 - AlmaLinux, Amazon Linux 2, Amazon Linux 2023, CentOS Stream 9, FreeBSD, and openSUSE instances: `ec2-user`
 - CentOS 7 instances: `centos`
 - Debian instances: `admin`
 - Ubuntu instances: `ubuntu`
 - Bitnami instances: `bitnami`

- Plesk instances: ubuntu
- cPanel & WHM instances: centos

For more information about instance operating systems, see [Choosing an image in Lightsail](#).

To learn more about SSH, see [SSH and connecting to your Amazon Lightsail instance](#).

Connect to your Lightsail Linux instance using SFTP

You can transfer files between your local computer and your Linux or Unix instance in Amazon Lightsail by connecting to your instance using SFTP (SSH File Transfer Protocol). To do this, you must get the private key for your instance, and then use it to configure the FTP client. This tutorial shows you how to configure the FileZilla FTP client to connect to your instance. These steps may also apply to other FTP clients.

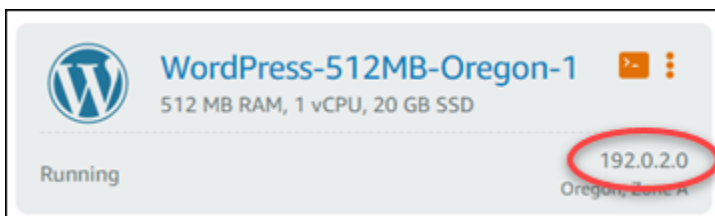
Contents

- [Prerequisites](#)
- [Get the SSH key for your instance](#)
- [Configure FileZilla and connect to your instance](#)

Prerequisites

Complete the following prerequisites if you haven't already:

- Download and install FileZilla on your local computer. For more information, see the following download options:
 - [Download FileZilla Client for Windows](#)
 - [Download FileZilla Client for Mac OS X](#)
 - [Download FileZilla Client for Linux](#)
- Get the public IP address of your instance. Sign in to the [Lightsail console](#), and then copy the public IP address that is displayed next to your instance, as shown in the following example:



Get the SSH key for your instance

Complete the following steps to get the default private key for the AWS Region of your instance, which is required to connect to your instance using FileZilla.

Note

If you're using your own key pair, or you created a key pair using the Lightsail console, locate your own private key and use it to connect to your instance. Lightsail does not store your private key when you upload your own key or create a key pair using the Lightsail console. You cannot connect to your instance using SFTP without your private key.











1. Sign in to the [Lightsail console](#).
2. Choose **Account** on the top navigation bar, and then choose **Account** from the drop-down.
3. Choose the **SSH Keys** tab.
4. Scroll down to the **Default keys** section of the page.
5. Choose **Download** next to the default private key for the region where your instance is located.

Default keys

With default key pairs, you can connect to Linux instances using an SSH client, and retrieve administrator passwords for Windows instances.

You can download or delete your default key pairs. You can create one default key per AWS Region where you previously created resources.

[+ Create key pair](#)

Region name	Region code	Created		
 Frankfurt	eu-central-1	April 27, 2018, 3:14 PM		
 Ireland	eu-west-1	April 27, 2018, 3:14 PM		
 Mumbai	ap-south-1	April 20, 2018, 2:54 PM		
 Ohio	us-east-2	February 2, 2022, 4:17 PM		
 Oregon	us-west-2	April 19, 2018, 9:11 AM		
 Seoul	ap-northeast-2	August 23, 2018, 9:11 AM		
 Singapore	ap-southeast-1	June 20, 2018, 3:45 PM		
 Stockholm	eu-north-1	May 13, 2021, 10:03 AM		
 Sydney	ap-southeast-2	April 30, 2019, 3:51 PM		

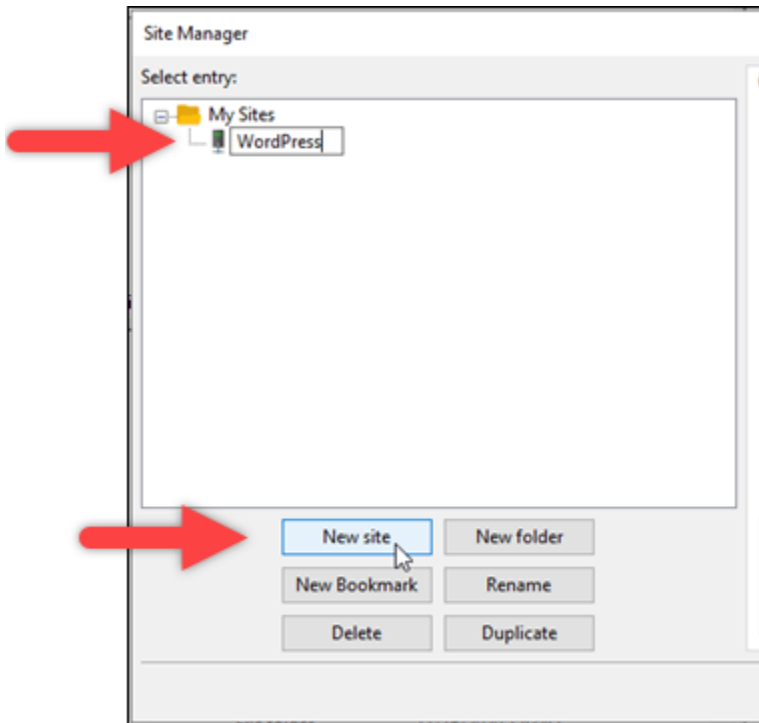


6. Save your private key in a secured location on your local drive.

Configure FileZilla and connect to your instance

Complete the following steps to configure FileZilla to connect to your instance.

1. Open FileZilla.
2. Choose **File, Site Manager**.
3. Choose **New site**, then give your site a name.



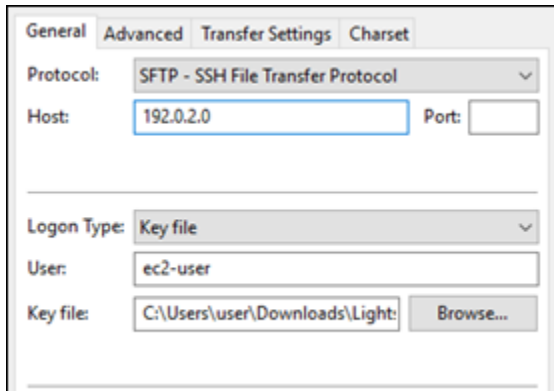
4. In the **Protocol** dropdown, choose **SFTP – SSH File Transfer Protocol**.
5. In the **Host** text box, enter or paste your instance's public IP address.
6. In the **Logon Type** dropdown, choose **Key File**.
7. In the **User** text box, enter one of the following default user names depending on your instance operating system:
 - AlmaLinux, Amazon Linux 2, Amazon Linux 2023, CentOS Stream 9, FreeBSD, and openSUSE instances: `ec2-user`
 - CentOS 7 instances: `centos`
 - Debian instances: `admin`
 - Ubuntu instances: `ubuntu`
 - Bitnami instances: `bitnami`
 - Plesk instances: `ubuntu`

- cPanel & WHM instances: centos

Important

If you are using a different user name than the default user names listed here, then you might need to give the user write permissions to your instance.

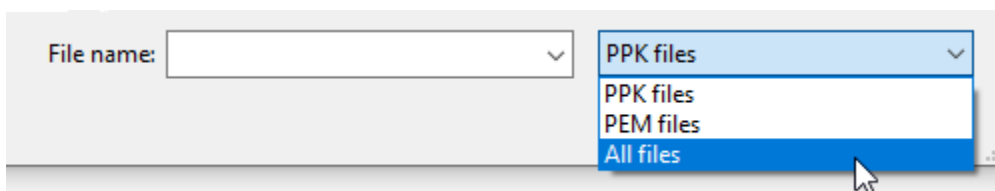
8. Next to the **Key File** text box, choose **Browse**.



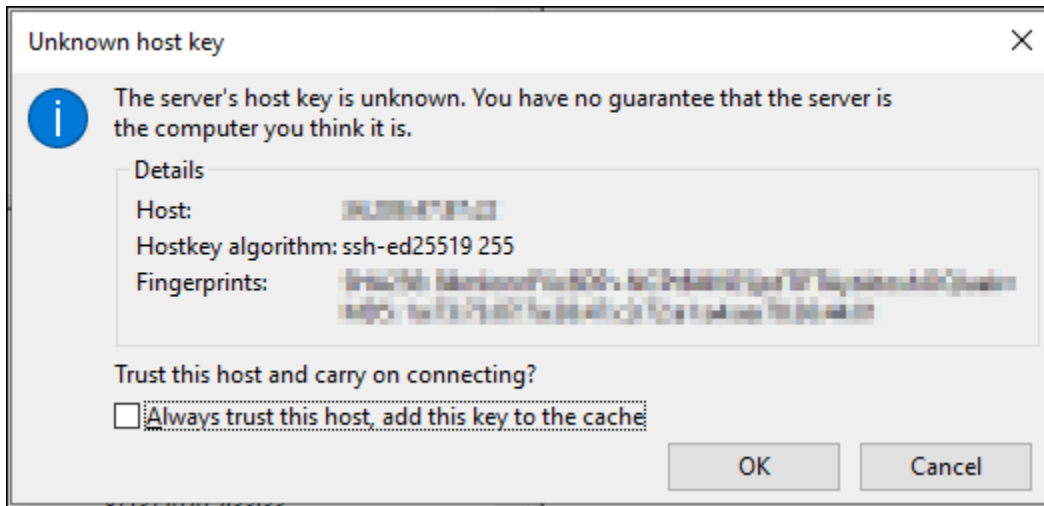
9. Locate the private key file that you downloaded from the Lightsail console earlier in this procedure, and then choose **Open**.

Note

If you are using Windows, change the default file type to **All files** when searching for your pem file.



10. Choose **Connect**.
11. You may see a prompt similar to the following example, indicating that the host key is unknown. Choose **OK** to acknowledge the prompt and connect to your instance.



You are successfully connected if you see status messages similar to the following example:

```
Status:      Connecting to 192.0.2.0. .
Status:      Connected to 192.0.2.0
Status:      Retrieving directory listing...
Status:      Listing directory /home/ec2-user
Status:      Directory listing of "/home/ec2-user" successful
```

For more information about using FileZilla, including how to transfer files between your local computer and your instance, see the [FileZilla Wiki page](#).

Manage SSH keys in Amazon Lightsail

You can establish a secure connection to your Amazon Lightsail instances using key pairs. When you first create an Amazon Lightsail instance, you can choose to use a key pair that Lightsail creates for you (the Lightsail default key pair) or a custom key pair that you create. For more information, see [Key pairs and connecting to instances in Amazon Lightsail](#).

On Linux and Unix instances, the private key allows you to establish a secure SSH connection to your instance. On Windows instances, the private key decrypts the default administrator password that you use to establish a secure RDP connection to your instance.

In this guide, we show you how to manage the keys that you can use with your Lightsail instances. You can view your keys, delete existing keys, and create or upload new keys.

Contents

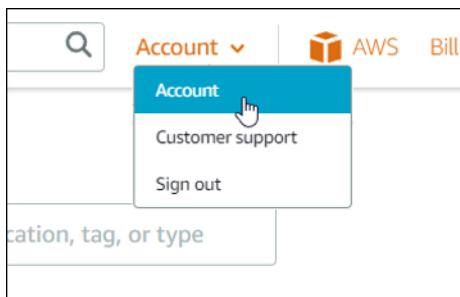
- [View your default and custom keys](#)

- [Download the private key of a default key from the Lightsail console](#)
- [Delete a custom key in the Lightsail console](#)
- [Delete a default key and create a new one in the Lightsail console](#)
- [Create a custom key using the Lightsail console](#)
- [Create a custom key using ssh-keygen and upload to Lightsail](#)

View your default and custom keys

Complete the following procedure to view your default and custom keys from the Lightsail console.

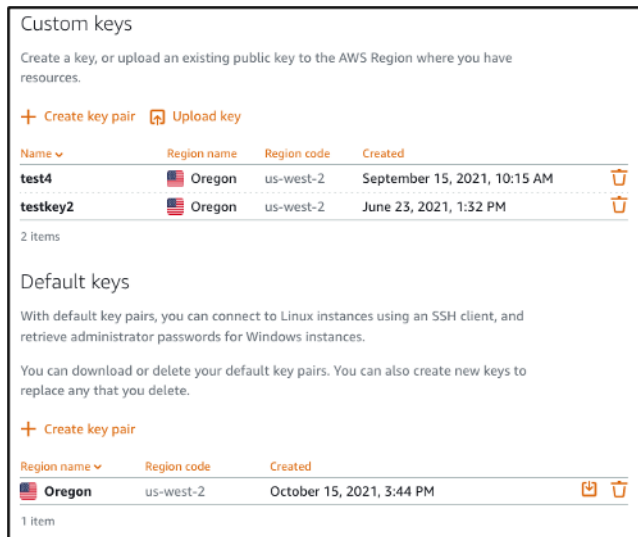
1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose **Account** on the top navigation menu.
3. Choose **Account** in the dropdown menu.



4. Choose the **SSH keys** tab.

The **SSH keys** page lists:

- **Custom keys** – These are keys that you create either using the Lightsail console or a third-party tool such as ssh-keygen. You can have many custom keys in each AWS Region.
- **Default keys** – These are keys that Lightsail creates for you. You can have only one default key in each AWS Region.



The screenshot shows the 'Custom keys' section of the Amazon Lightsail console. It includes a header 'Custom keys' and a sub-header 'Create a key, or upload an existing public key to the AWS Region where you have resources.' Below this are two buttons: '+ Create key pair' and 'Upload key'. A table lists two custom keys:

Name	Region name	Region code	Created	
test4	Oregon	us-west-2	September 15, 2021, 10:15 AM	
testkey2	Oregon	us-west-2	June 23, 2021, 1:32 PM	

Below the table, it says '2 Items'. The 'Default keys' section follows, with a sub-header 'Default keys' and a description: 'With default key pairs, you can connect to Linux instances using an SSH client, and retrieve administrator passwords for Windows instances. You can download or delete your default key pairs. You can also create new keys to replace any that you delete.' There is a '+ Create key pair' button and a table with one default key:

Region name	Region code	Created	
Oregon	us-west-2	October 15, 2021, 3:44 PM	

Below the table, it says '1 Item'.

Custom and default keys are Regional. For example, keys in the US West (Oregon) AWS Region can be configured only on instances created in that Region. For more information about keys, see [Key pairs and connecting to instances in Amazon Lightsail](#).

On the **SSH keys** page, you can create key pairs, upload keys, delete keys, and download the private key of a Lightsail default key pair.

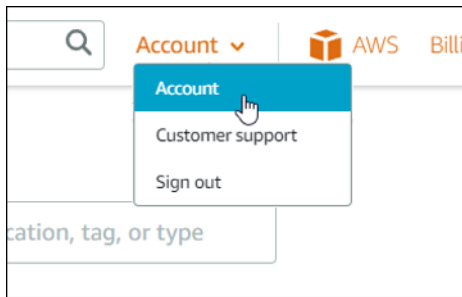
Note

You cannot download the private key of a custom key pair because Lightsail does not store that key for you. If you've lost the private key of a custom key pair, then you should create a new one, and configure it on your instance. Then, delete the key which has been lost. For more information, see [Create a custom key using the Lightsail console](#) or [Create a custom key using ssh-keygen and upload to Lightsail](#) later in this guide.

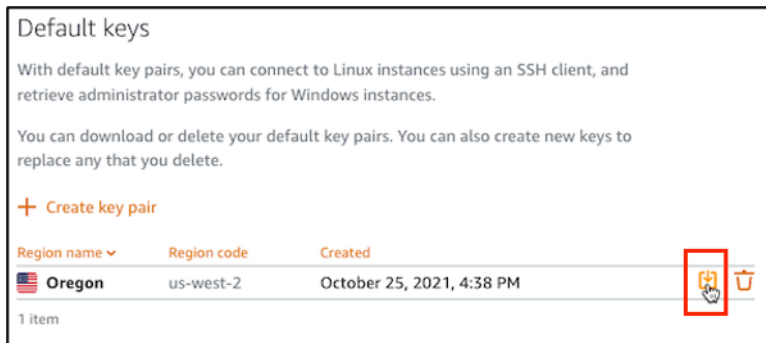
Download the private key of a default key from the Lightsail console

Complete the following procedure to download the private key of a default key pair from the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose **Account** on the top navigation pane.
3. Choose **Account** in the dropdown menu.



4. Choose the **SSH keys** tab.
5. Under the **Default keys** section of the page, choose the download icon for the key that you want to download.



Important

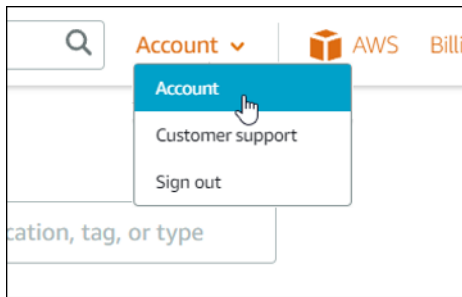
Store the private key in a secure location. Don't share it publicly because it can be used to connect to your instances.

You can configure an SSH client to connect to your instances using the private key. For more information, see [Connecting to your instances](#).

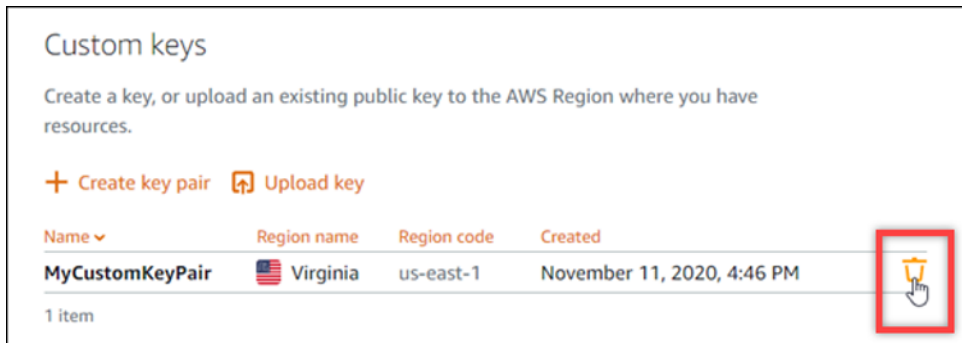
Delete a custom key in the Lightsail console

Complete the following procedure to delete a custom key in the Lightsail console. This prevents the custom key from being configured on new instances that you create in Lightsail.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose **Account** on the top navigation pane.
3. Choose **Account** in the dropdown menu.



4. Choose the **SSH keys** tab.
5. Under the **Custom keys** section of the page, choose the delete icon for the key that you want to delete.

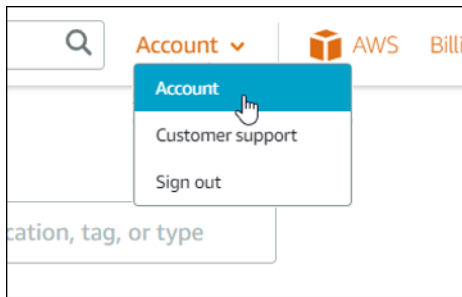


This doesn't remove the public key of the custom key pair from instances that were previously created and are currently running. To remove a previously configured public key stored on a running instance, see [Manage keys stored on an instance in Amazon Lightsail](#).

Delete a default key and create a new one in the Lightsail console

Complete the following procedure to delete a default key in the Lightsail console. This prevents that default key from being configured on new instances that you create in Lightsail. You can then create a new default key to replace the one that you deleted. You will be able to configure the new default key on new instances that you create in Lightsail.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail homepage, choose **Account** on the top navigation pane.
3. Choose **Account** in the dropdown menu.



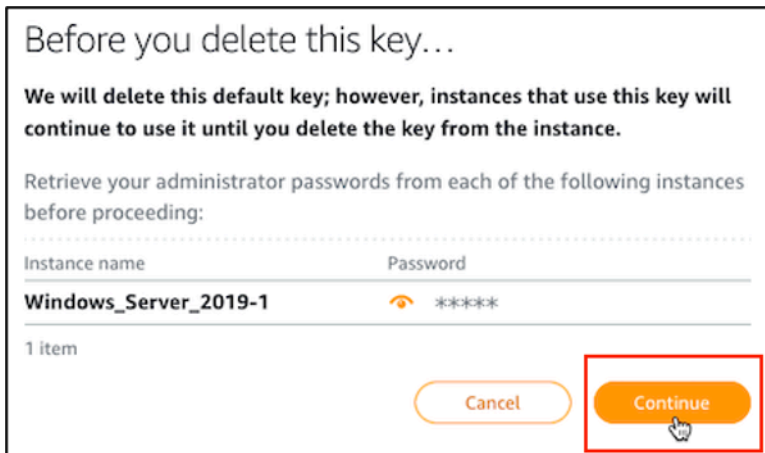
4. Choose the **SSH keys** tab.
5. Under the **Default keys** section of the page, choose the delete icon for the default key that you want to delete.



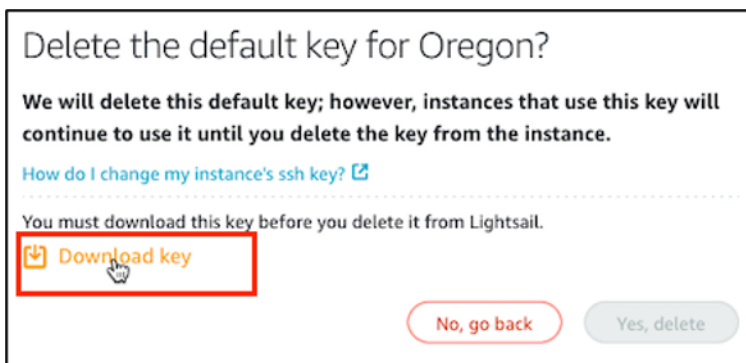
Important

Deleting a default key doesn't remove the public key of the custom key pair from instances that were previously created and are currently running. For more information, see [Manage keys stored on an instance in Amazon Lightsail](#).

6. The default key is used to generate the administrator password for Windows instances. Before you delete the default key, you should retrieve and save the administrator password from any Windows instances that use the default key you want to delete.
7. Choose **Continue** to delete the default key.



- You must download the default key before you can delete it. After you download the default key, you will be able to choose **Yes, delete** to permanently delete the default key.



- The default key has been deleted. Choose **Okay**.



The following steps are optional and you should only complete them if you want to replace the default key pair you deleted.

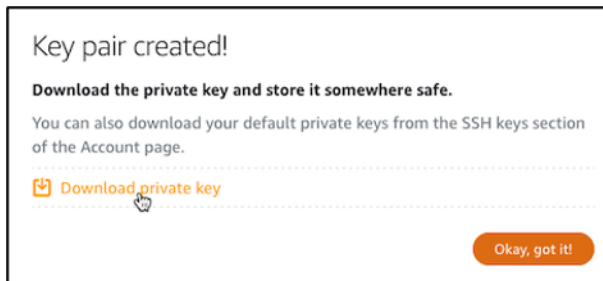
- Under the **Default keys** section of the page, choose **Create key pair**.
- In the **Select a region** prompt that appears, choose the AWS Region in which you want to create your new default key. You will be able to configure your new default key on new instances in the same AWS Region.

Note

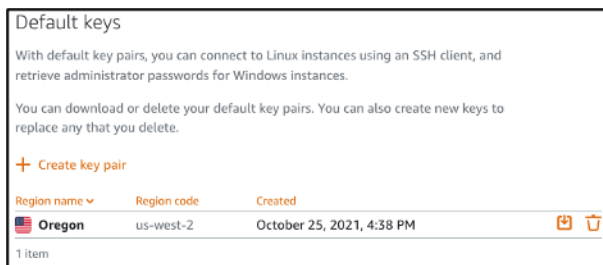
Using these steps, you can create default key pairs only in AWS Regions where you have created Lightsail resources. To create a default key pair in a new Region, you must create a Lightsail resource in that Region. Creating the resource also creates a default key pair.

12. Download the private key and store it in a safe location.

13. Choose **Ok, got it!** to continue.



14. Confirm the new default key on the Lightsail console SSH keys page.

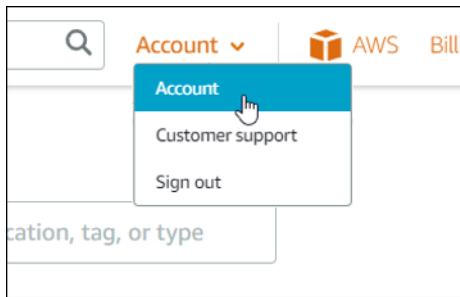


You can configure your new default key on new instances that you create in Lightsail. To configure your new default key on instances that were previously created and are currently running, see [Manage keys stored on an instance in Amazon Lightsail](#).

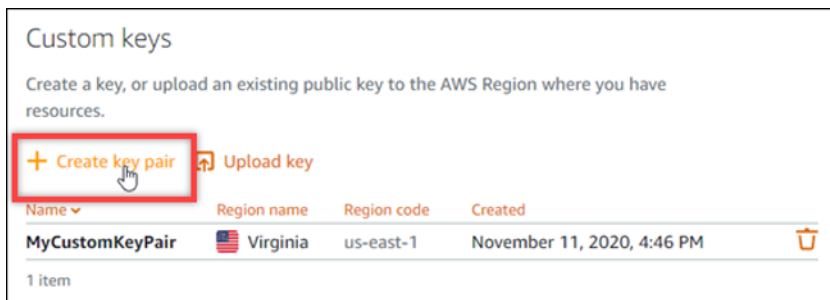
Create a custom key using the Lightsail console

Complete the following procedure to create a custom key pair using the Lightsail console. You will be able to configure the new custom key on new instances that you create in Lightsail.

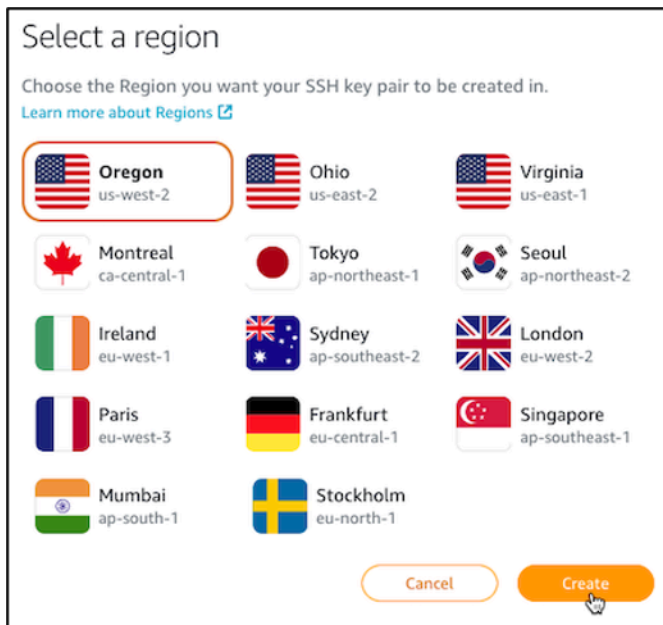
1. Sign in to the [Lightsail console](#).
2. On the Lightsail homepage, choose **Account** on the top navigation pane.
3. Choose **Account** in the dropdown menu.



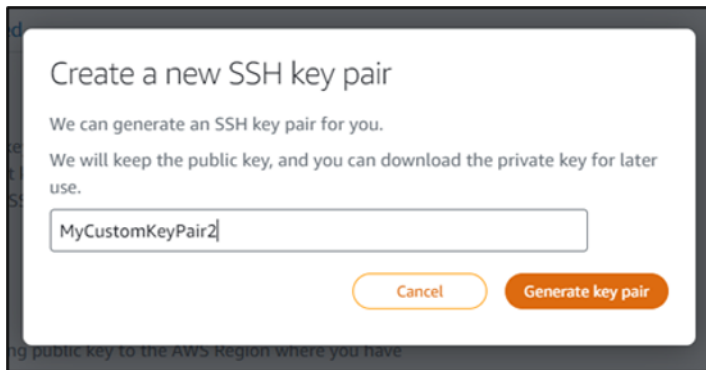
4. Choose the **SSH keys** tab.
5. Choose **Create key pair** under the **Custom keys** section of the page.



6. In the **Select a region** prompt that appears, choose the AWS Region in which you want to create your new custom key. You will be able to configure your new custom key on new instances in the same AWS Region.



7. In the **Create a new SSH key pair** prompt that appears, give your custom key a name, and choose **Generate key pair**.

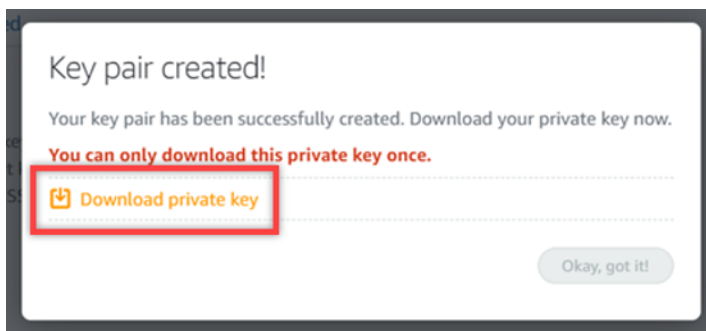


8. In the **Key pair created!** prompt that appears, choose **Download private key** to save the private key to your local computer.

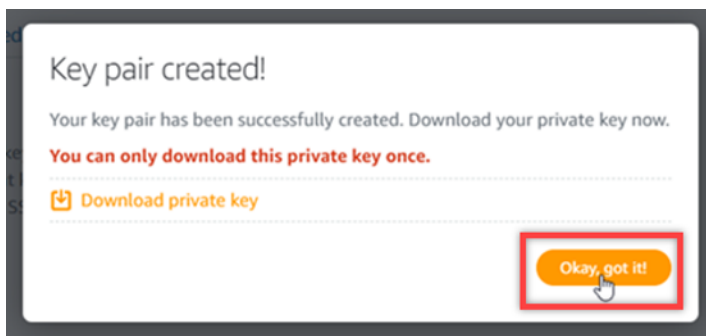
⚠ Important

Store the private key in a secured location. Don't share it publicly because it can be used to connect to your instances.

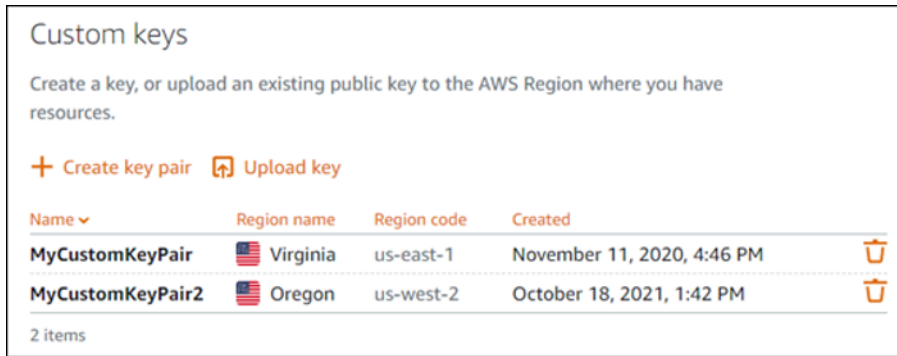
This is the only time you can download the private key of the custom key pair. Lightsail does not store the private key of custom key pairs. After you close this prompt, you will not be able to download it again.



9. Choose **Ok, got it!** to close the prompt.



10. Your new custom key is listed under the Custom keys section of the page.



You can configure your new custom key on new instances that you create in Lightsail. To configure your new custom key on instances that were previously created and are currently running, see [Manage keys stored on an instance in Amazon Lightsail](#).

Create a custom key using ssh-keygen and upload to Lightsail

Complete the following procedure to create a custom key pair on your local computer using a third-party tool, such as ssh-keygen. After you create the key, you can upload it to the Lightsail console. You will be able to configure the new custom key on new instances that you create in Lightsail.

1. Open Command Prompt or Terminal on your local computer.
2. Enter the following command to create a key pair.

```
ssh-keygen -t rsa
```

3. Specify a directory location on your computer where the key pair should be saved.

For example, you can specify one of the following directories:

- a. On Windows: `C:\Users\<UserName>\.ssh\<KeyPairName>`
- b. On macOS, Linux or Unix: `/home/<UserName>/.ssh/<KeyPairName>`

Replace *<UserName>* with the name of the user you're currently signed in as, and replace *<KeyPairName>* with the name of your new key pair.

In the following example, we specified the `C:\Keys` directory on our Windows computer, and gave the new key a name of `MyNewLightsailCustomKey`.

```
C:\Users\██████>ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\██████/.ssh/id_rsa): C:\Keys\MyNewLighstailCustomKey
```

4. Enter a passphrase for your key and press **Enter**. You will not see the passphrase as you enter it.

You will need this passphrase later when configuring the private key of the key pair on an SSH client to connect to an instance that has the public key of the key pair configured on it.

```
Enter passphrase (empty for no passphrase):
```

5. Enter the passphrase again to confirm it and press **Enter**. You will not see the passphrase as you enter it.

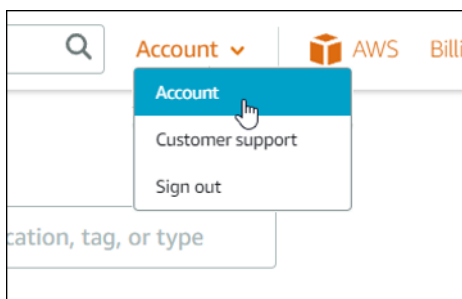
```
Enter same passphrase again:
```

6. A prompt confirms that your private key and public key have been saved to the specified directory.

```
Your identification has been saved in C:\Keys\MyNewLighstailCustomKey.
Your public key has been saved in C:\Keys\MyNewLighstailCustomKey.pub.
```

Next you will upload the public key of the key pair to the Lightsail console.

7. Sign in to the [Lightsail console](#).
8. On the Lightsail home page, choose **Account** on the top navigation pane.
9. Choose **Account** in the dropdown menu.



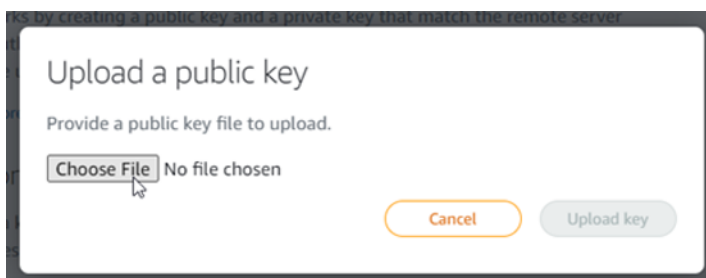
10. Choose the **SSH keys** tab.
11. Choose **Upload key** under the **Custom keys** section of the page.



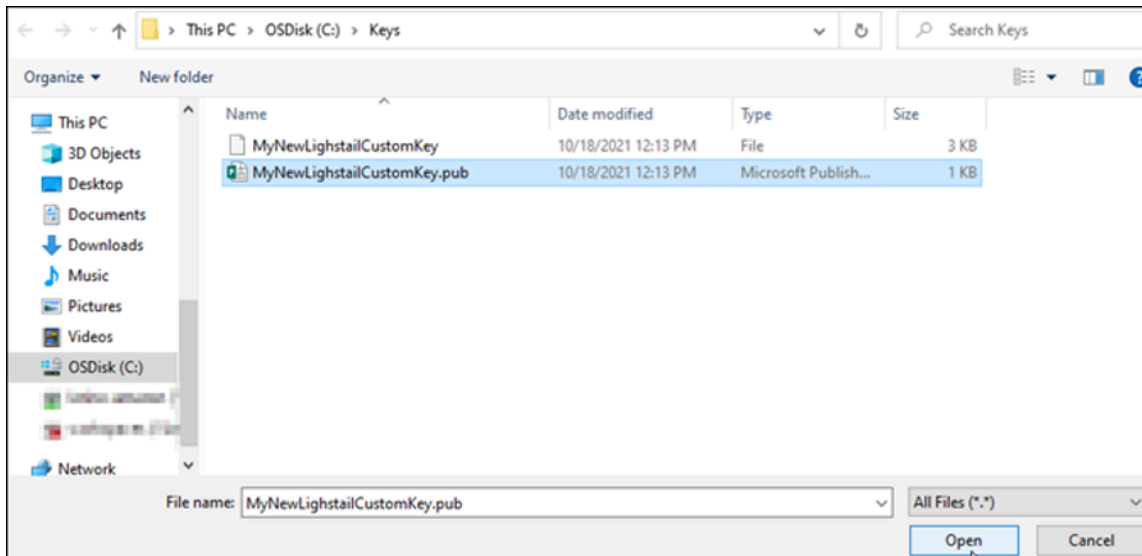
- In the **Select a region** prompt that appears, choose the AWS Region in which you want to upload your new custom key. You will be able to configure your new custom key on new instances in the same AWS Region.



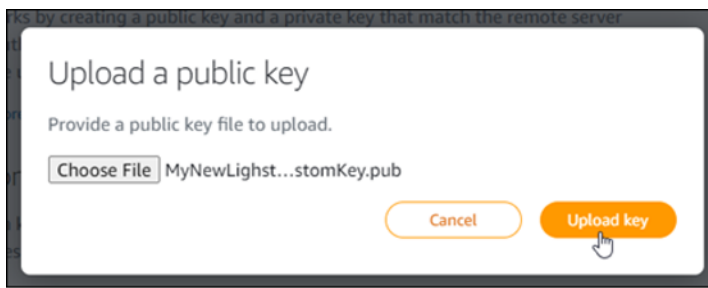
- Choose **Upload**.
- Click **Choose File** in the **Upload a public key** prompt that appears.



- Find the public key of the key pair you created earlier in this procedure, on your local computer, and choose **Open**. The public key of the key pair is the file with a .PUB file extension.



16. Choose **Upload key**.



17. Your new custom key is listed in the **Custom keys** section of the page.



You can configure your new custom key on new instances that you create in the AWS Region where you uploaded your key. To configure your new custom key on instances that were previously created and are currently running, see [Manage keys stored on an instance in Amazon Lightsail](#).

Manage SSH keys stored on a Lightsail instance

You can establish a secure connection to your Amazon Lightsail instances using key pairs. Lightsail configures the public key of a key pair on your Linux or Unix instance when you first create it. You use the private key of the key pair to authenticate to your instance when establishing an SSH connection to it. For more information about keys, see [Key pairs and connecting to instances](#).

After your instance is up and running, you can change the key pair that is used to connect to your instance by adding a new public key on the instance, or by replacing the public key (deleting the existing public key and adding a new one) on the instance. You might do this for the following reasons:

- If a user in your organization requires access to the instance using a separate key pair, you can add the public key to your instance.
- If you need to secure a new instance that was created from the snapshot of an instance which used a compromised key.
- If someone has a copy of the private key and you want to prevent them from connecting to your instance (for example, if they left your organization), you can delete the public key on the instance and replace it with a new one.

To add or replace a key on your instance, you must be able to connect to your instance. If you've lost your existing private key, you can connect to your instance using the Lightsail browser-based SSH client. For more information, see [Connect to your Linux or Unix instance](#).

Contents

- Step 1: [Learn about the process](#)
- Step 2: [Create a key pair](#)
- Step 3: [Add a public key to your instance](#)
- Step 4: [Connect to your instance using the new key pair](#)
- Step 5: [Delete an existing public key from your instance](#)

Step 1: Learn about the process

Following are the general steps to add and remove keys on an instance. If you want to remove a key from your instance without adding a new key, see Step 5: [Delete an existing public key from your instance](#) later in this guide.

1. **Create a key pair** – To add a new key to your instance you must first create a new key pair. You can create a custom or default key pair using the Lightsail console, or on your local computer using a third-party tool, such as ssh-keygen. Both methods generate a new key pair, which consist of a public key and a private key. For more information, see Step 2: [Create a key pair](#) later in this guide.
2. **Add a public key to your instance** – After you create a key pair, you connect to your instance using SSH and add the public key of the key pair to your instance. For more information, see Step 3: [Add a public key to your instance](#) later in this guide.
3. **Test that you can connect to your instance using the new key pair** – After the public key of the key pair is saved on the instance, you should test that you can use the private key of the key pair to connect to the instance using SSH. For more information, see Step 4: [Connect to your instance using the new key pair](#) later in this guide.
4. **Remove an old public key from your instance** – After you successfully connect to your instance using the new key, you can remove an old public key from the instance. Complete this step to prevent a user from connecting to an instance using an old key pair. For more information, see Step 5: [Delete an existing public key from your instance](#) later in this guide.

Step 2: Create a key pair

Complete the following procedure to create a key pair on your local computer using ssh-keygen.

1. Open Command Prompt or Terminal on your local computer.
2. Enter the following command to create a key pair.

```
ssh-keygen -t rsa
```

3. Specify a directory location on your computer where the key pair should be saved.

For example:

- On Windows: C:\Users*<UserName>*\.ssh*<KeyPairName>*
- On macOS, Linux or Unix: /home/*<UserName>*/.ssh/*<KeyPairName>*

Replace *<UserName>* with the name of the user you are currently signed in as, and replace *<KeyPairName>* with the name of your new key pair.

In the following example, we specified the C:\Keys directory on our Windows computer, and gave the new key a name of MyNewLightsailCustomKey.

```
C:\Users\<user>>ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\<user>\.ssh/id_rsa): C:\Keys\MyNewLighstailCustomKey
```

4. Enter a passphrase for your key and press **Enter**. You will not see the passphrase as you enter it.

You will need this passphrase later when configuring the private key on an SSH client to connect to an instance that has the public key configured on it.

```
Enter passphrase (empty for no passphrase):
```

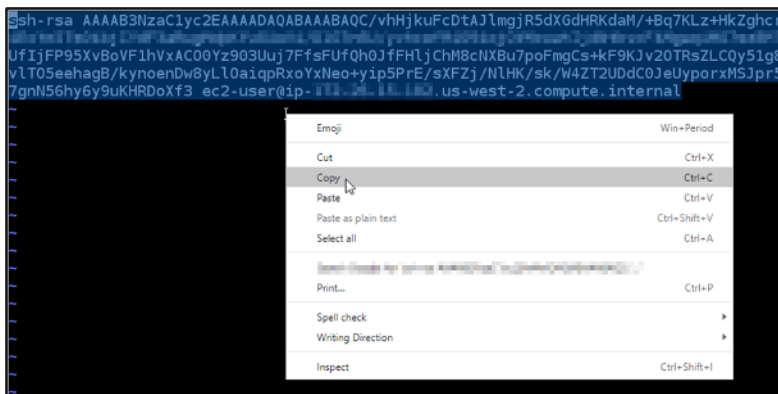
5. Enter the passphrase again to confirm it and press **Enter**. You will not see the passphrase as you enter it.

```
Enter same passphrase again:
```

6. A prompt confirms that your private key and public key have been saved to the specified directory.

```
Your identification has been saved in C:\Keys\MyNewLighstailCustomKey.
Your public key has been saved in C:\Keys\MyNewLighstailCustomKey.pub.
```

7. Open the public key (.PUB) file, and copy the text in the file.



Continue to the next section of this guide to add your new public key to your Lightsail instance.

Step 3: Add a public key to your instance

Complete the following procedure to add the public key to your instance. Public key content is saved in the `~/.ssh/authorized_keys` file on Linux and Unix instances.

1. Sign in to the [Lightsail console](#).
2. Choose the **Instances** tab on the Lightsail home page.
3. Choose the browser-based SSH client icon for the instance that you want to connect to.



4. After you're connected, enter the following command to edit the `authorized_keys` file using the text editor of your choice. The following steps use Vim for demonstration purposes.

```
sudo vim ~/.ssh/authorized_keys
```

You should see a result similar to the following example, which shows the current public keys configured on your instance. In our case, the Lightsail default key for the AWS Region in which the instance was created in, is the only public key configured on the instance.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC+QizYnwmJ.
R6b23qBWH00Siy5uUFh5YYn4TX5I5Q70cIA+l5AGxjZpWiyRl
dFL5RwR1Dws7pret5LC6l+PSalD4eJ7g2z0RUKIf6G6G1NehI
vyXdzVeg0GQiflMbez0V LightsailDefaultKeyPair
~
~
~
```

5. Press the **I** key to enter insert mode in the Vim editor.
6. Enter a line break after the last public key on the file.
7. Paste the public key text that you copied earlier in this guide (after creating a new key pair). You should see a result similar to the following example:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC+QizYnwmJZ63wmRgTWSlkI7gF0qQl4sqIf5Z2
R6b23qBWH00Siy5uUFh5Yyn4TX5I5Q70cIA+l5AGxjZpWiyRBo5YFBgSP00T0wR9A+s55DYU6rSY
dFL5RwR1Dws7pret5LC6l+PSaLD4eJ7g2z0RUkIf6G6G1NehLmupFYqaPPiEV8DAtWSj qoHgEaj9
vyXdzVeg0G0iflMbez0V LightsailDefaultKeyPair
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC/vhHjkuFcDtAJlmgjR5dXGdHRKdaM/+Bq7KLz
UfIjFP95XvBoVF1hVxAC00Yz903Uuj7FfsFUfQh0JfFHljChM8cNXBu7poFmgCs+kF9KJv20TRsZ
v1T05eehagB/kynoenDw8yL10aiqpRxoYxNeo+yip5PrE/sXFZj/NLHK/sk/W4ZT2UDdC0JeUyp0
7gnN56hy6y9uKHRDoXf3 ec2-user@ip-10-10-10-10.us-west-2.compute.internal
```

8. Press the **ESC** key. Next, type `:wq!` and press **Enter** to save your edits and exit the Vim editor.

The new public key is now added to your instance. Continue to the next section of this guide to connect to your instance using the new key pair.

Step 4: Connect to your instance using the new key pair

To test the new key pair, disconnect from your instance, and reconnect to it using the private key that you created earlier in this guide. For more information, see [Key pairs and connecting to instances in Amazon Lightsail](#). After you successfully connect to your instance using the new key, you can remove an old key from the instance. Continue to the next step to learn how to delete public keys from your instance.

Step 5: Delete an existing public key from your instance

Complete the following procedure to remove a public key from your instance. This prevents a user from connecting to an instance using an old key pair. Do this after you successfully connect to the instance using the new key pair.

1. Connect to your instance using SSH.
2. Enter the following command to edit the `authorized_keys` file using the text editor of your choice. The following steps use Vim for demonstration purposes.

```
sudo vim ~/.ssh/authorized_keys
```

3. Press the letter **I** key to enter insert mode in the Vim editor.
4. Delete the line of text that contains the public key that you want to remove from your instance.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC+QizYnwmJZ63wmRgTWSlkI7gF0qQl4sqIf5Z
RgB23qBWH00Siy5uUFh5YYn4TX5I5Q70cIA+l5AGxj2pWlyK25YERqSP0QT0wR9A+s55DYU6rS
dFL5RwR1Dws7pret5LC6l+PSa7B4eJ7g2Z0RUkIf6G6G1NehLmupFYqaPP1EV8DAthSjqHqFaj
vvXdzVsq00uITLMbez0V LightsailDefaultKeyPair
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC/vhHjkuFcDtAJlmgjR5dXGdHRKdaM/+Bq7KL
UfIjFP95XvBoVF1hVxAC00Yz903Uuj7FfsFUfQh0JfFHljChM8cNXBu7poFmgCs+kF9KJv20TRs
vLT05eehagB/kynoenDw8yLl0aiqpRxoYxNeo+yip5PrE/sXFZj/NlHK/sk/W4ZT2UDdC0JeUyp
7gnN56hy6y9uKHRDoXf3 ec2-user@ip-10.0.0.1.us-west-2.compute.internal
~
~
```

The result should look like the following example, where the new public key is the only key that displays.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC/vhHjkuFcDtAJlmgjR5dXGdHRKdaM/+Bq7KL
UfIjFP95XvBoVF1hVxAC00Yz903Uuj7FfsFUfQh0JfFHljChM8cNXBu7poFmgCs+kF9KJv20TRs
vLT05eehagB/kynoenDw8yLl0aiqpRxoYxNeo+yip5PrE/sXFZj/NlHK/sk/W4ZT2UDdC0JeUyp
7gnN56hy6y9uKHRDoXf3 ec2-user@ip-10.0.0.1.us-west-2.compute.internal
~
~
```

5. Press the **ESC** key. Next, type `:wq!` and press **Enter** to save your edits and exit the Vim editor.

The deleted public key is now removed from your instance. Your instance will refuse connections that use the private key of that key pair.

Download and set up PuTTY for Lightsail

You can use an SSH client like PuTTY to connect to your Lightsail instance. PuTTY requires a copy of your private SSH key. You might already have a key, or you might want to use the key pair that Lightsail creates. Either way, we've got you covered. For more information about SSH, see [SSH key pairs](#). This topic walks you through the steps to download a key pair and set up PuTTY to connect to your instance.

The method to connect to your instance described in this guide is one of many. For more information about the other methods, see [SSH key pairs](#).

The easiest way to connect to your Linux or Unix instance in Lightsail is by using the browser-based SSH client that is available in the Lightsail console. For more information, see [Connecting to your Linux or Unix instance in Amazon Lightsail](#).

Prerequisites

- You need a running instance in Lightsail. For more information, see [Create an instance in Amazon Lightsail](#).
- We recommended that you create a static IP address and attach it to your instance so you won't have to reconfigure PuTTY if your public IP address changes later. For more information, see [Create a static IP and attach it to an instance](#).

Step 1: Download and install PuTTY

PuTTY is a free implementation of SSH for Windows. Learn more about PuTTY on the [PuTTY website](#), including restrictions related to countries where encryption isn't allowed. If you already have PuTTY, you can skip to **Step 2**.

1. Download the PuTTY installer or executable file from the following link: [Download PuTTY](#).

If you need help deciding which download to choose, see the [PuTTY documentation](#). We recommend using the latest version.

2. Go on to **Step 2** to get your private key before you configure PuTTY.

Step 2: Get your private key ready

You have several options for getting your private key. You might want to use the default private key that Lightsail generates, you might want to have Lightsail create a new private key for you, or you might already have one from another service. The steps for each of these options is outlined in the following procedures:

1. Sign in to the [Lightsail console](#).
2. Choose **Account** on the top navigation bar, and then choose **Account** from the drop-down.
3. Choose the **SSH Keys** tab.
4. Choose one of the following options depending on which private key you prefer to use:
 - **To use the default private key that Lightsail generates**, in the **Default keys** section of the page, choose the download icon next to the default private key for the AWS Region where your instance is located.

Default keys

With default key pairs, you can connect to Linux instances using an SSH client, and retrieve administrator passwords for Windows instances.

You can download or delete your default key pairs. You can create one default key per AWS Region where you previously created resources.

[+ Create key pair](#)

Region name	Region code	Created		
Frankfurt	eu-central-1	April 27, 2018, 3:14 PM		
Ireland	eu-west-1	April 27, 2018, 3:14 PM		
Mumbai	ap-south-1	April 20, 2018, 2:54 PM		
Ohio	us-east-2	February 2, 2022, 4:17 PM		
Oregon	us-west-2	April 19, 2018, 9:11 AM		
Seoul	ap-northeast-2	August 23, 2018, 9:11 AM		
Singapore	ap-southeast-1	June 20, 2018, 3:45 PM		
Stockholm	eu-north-1	May 13, 2021, 10:03 AM		
Sydney	ap-southeast-2	April 30, 2019, 3:51 PM		

- **To create a new key pair in Lightsail**, in the **Custom keys** section of the page, choose **Create key pair**. Choose the AWS Region where your instance is located, and choose **Create**. Enter a name, and choose **Generate key pair**. You will be given the option to download the private key.

Important

You can only download the private key once. Save it in a secured location.

- **To use your own key pair**, choose **Upload New**. Choose the AWS Region where your instance is located, and choose **Upload**. Choose **Upload file**, and then locate the file in your local drive. Choose **Upload key** when you're ready to upload your public key file to Lightsail.
5. If you downloaded the private key, or you created a new private key in Lightsail, then make sure to save the `.pem` key file somewhere you can easily find it.

We also recommend that you set permissions for the file so that no one else can read it.

Step 3: Configure PuTTYgen with your Lightsail private key

Now that you have a copy of your `.pem` key file, you can set up PuTTY using the PuTTY Key Generator (PuTTYgen).

1. Start PuTTYgen (for example, from the **Start** menu, choose **All Programs, PuTTY, PuTTYgen**).
2. Choose **Load**.

By default, PuTTYgen displays only files with the .ppk extension. To locate your .pem file, select the option to display files of all types.

3. Choose `lightsailDefaultKey.pem`, and then press **Open**.

PuTTYgen confirms that you successfully imported the key, and then you can choose **OK**.

4. Choose **Save private key**, and then confirm you don't want to save it with a passphrase.

If you choose to create a passphrase as an extra measure of security, remember you will need to enter it every time you connect to your instance using PuTTY.

5. Specify a name and a location to save your private key, and then choose **Save**.
6. Close PuTTYgen.

Step 4: Finish configuring PuTTY with your private key and instance information

You're almost there! Hang on while we make one last change.

1. Open PuTTY.
2. From Lightsail, grab the public IP address (hopefully you're using a [static IP address](#)) from the instance management page.

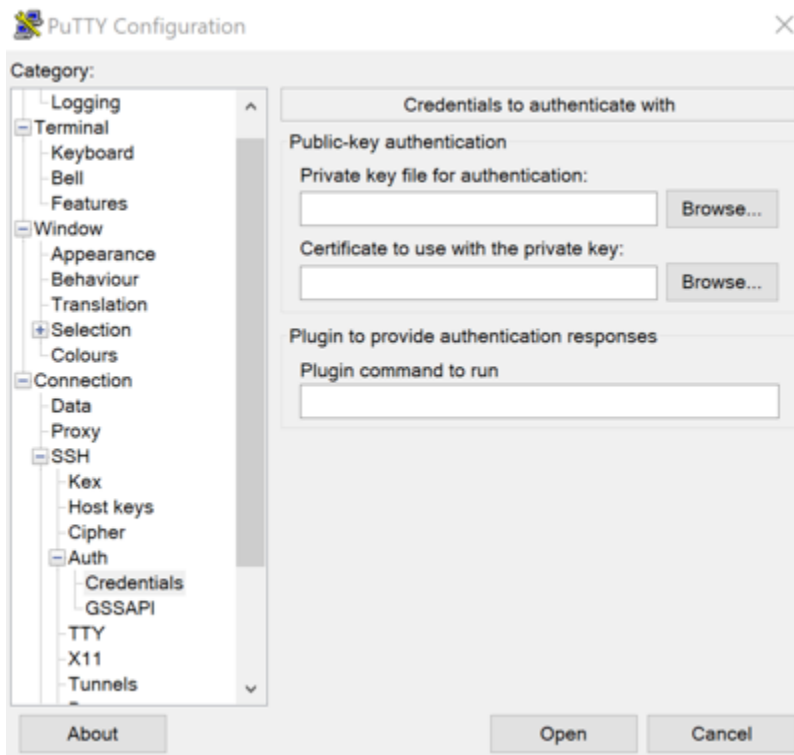
You can get the public IP address from the Lightsail home page, or choose your instance to view more details about it.

3. Type (or paste) the public IP address into the **Host Name (or IP address)** field.

Note

Port 22 is already open for SSH on your Lightsail instance, so accept the default port.

4. Under **Connection**, expand **SSH** and **Auth**, and then choose **Credentials**.



5. Choose **Browse** to navigate to the .ppk file that you created in the previous step, and then choose **Open**.
6. Choose **Open** again, and then choose **Accept** to trust this connection in the future.
7. Log in using one of the following default user names depending on your instance operating system:
 - AlmaLinux, Amazon Linux 2, Amazon Linux 2023, CentOS Stream 9, FreeBSD, and openSUSE instances: `ec2-user`
 - CentOS 7 instances: `centos`
 - Debian instances: `admin`
 - Ubuntu instances: `ubuntu`
 - Bitnami instances: `bitnami`
 - Plesk instances: `ubuntu`
 - cPanel & WHM instances: `centos`

For more information about instance operating systems, see [Choose an image](#).

8. Be sure to save your connection for future use.

Next steps

If you need to connect again, see [Connect to your Linux/Unix-based instance with PuTTY](#).

Connect to your Lightsail Windows instance

You can connect to your Windows Server instance in Amazon Lightsail using the browser-based RDP client that is available in the Lightsail console. The browser-based RDP client does not require software installation, and you can connect to your Windows Server instance immediately after you create it, and it becomes available. Connect to your instance to perform administrative tasks on the server, such as installing software, or configuring web applications.

Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

You can also use your own RDP client to connect to your instance, such as the Remote Desktop Connection that is bundled with Windows. For more information about configuring your own RDP client, see [Connect to your Windows instance with the Remote Desktop Connection client](#). To connect to a Linux or Unix instance in Lightsail, see [Connect to your Linux or Unix instance](#).

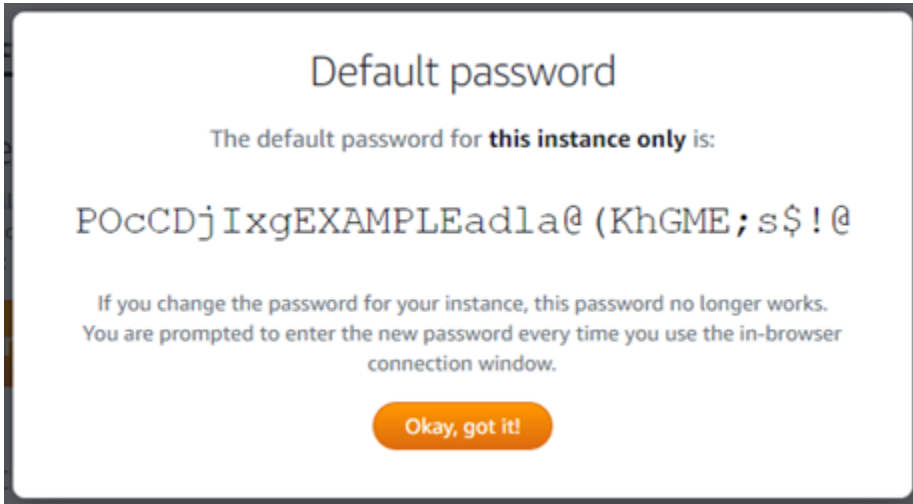
Default administrator password for Windows Server instances

A randomly generated default administrator password is assigned to Windows Server instances when they are created. The browser-based RDP client in the Lightsail console uses the default administrator password to sign in to your instance. If you change the administrator password on your instance, you will be prompted to manually enter your new password each time you try to connect to your instance using the browser-based RDP client. Lightsail does not store your new administrator password, and it cannot be retrieved from your instance.

Important

If you lose your administrator password, you will not be able to sign in to your instance, and there is no way to reset the password. Store your new administrator password in a secure location where you can retrieve it later if you lose it, such as AWS Secrets Manager. For more information, see the [AWS Secrets Manager User guide](#).

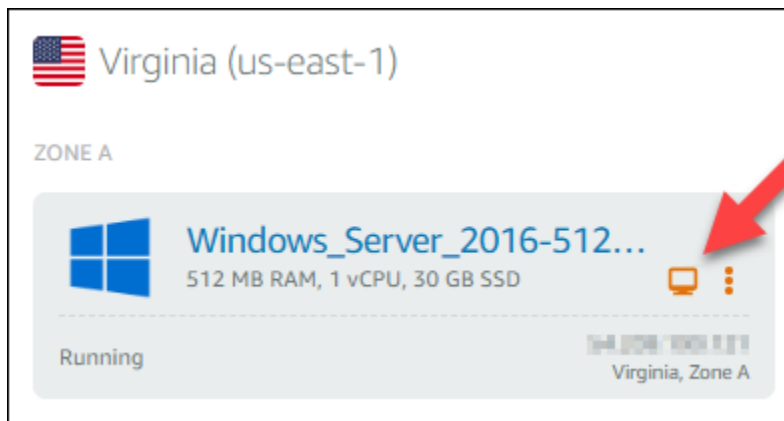
You can change the administrator password back to the original default administrator password to avoid being prompted for it each time you access your instance using the browser-based RDP client. You can find the original default administrator password by choosing the **Instances** tab in the [Lightsail home page](#). Choose the name of your Windows Server instance, choose the **Connect** tab, and choose **Show default password** to view the original default administrator password as shown in the following example.



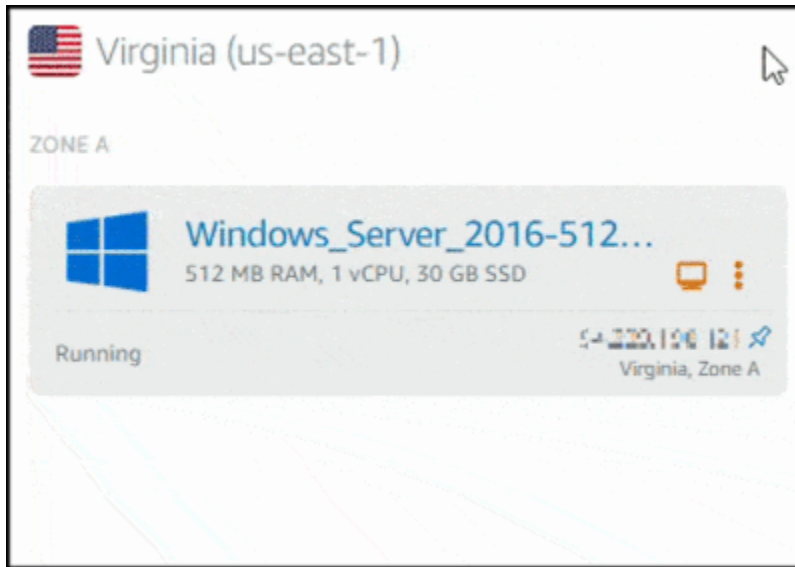
Connect to your Windows Server instance using the browser-based RDP client

Use the following procedure to connect to your Windows Server instance using the browser-based RDP client in the Lightsail console.

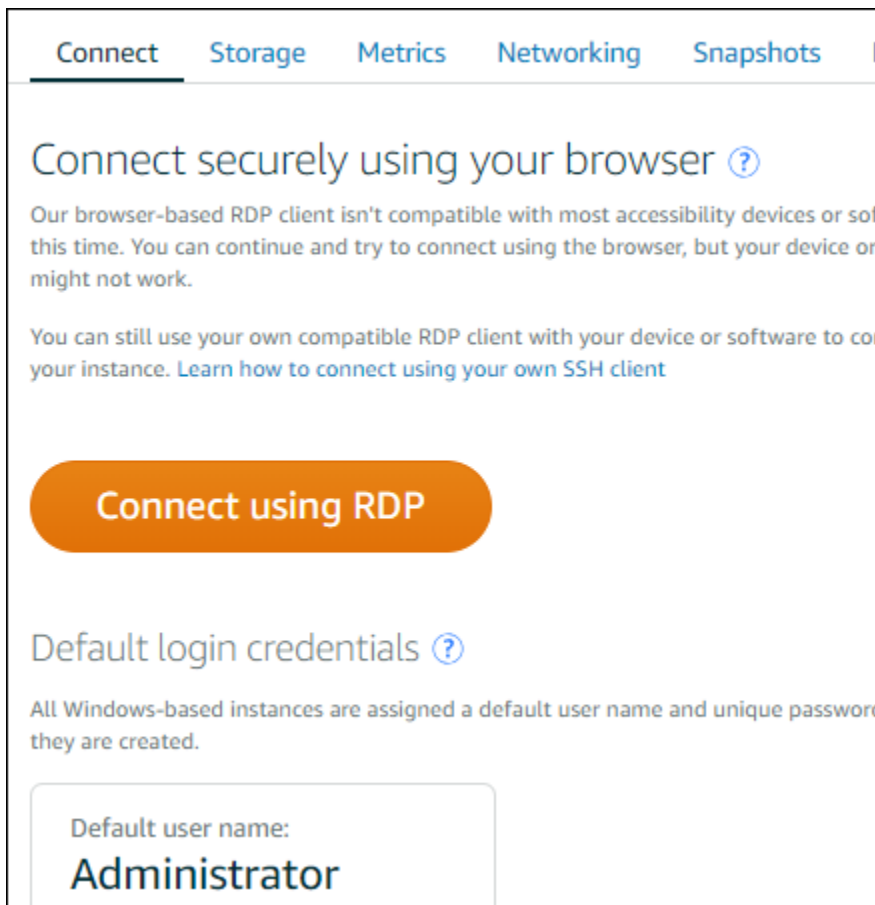
1. Sign in to the [Lightsail console](#).
2. Access the browser-based RDP client for the instance that you want to connect to by using one of the following steps:
 - Choose the browser-based RDP client icon, as shown in the following example.



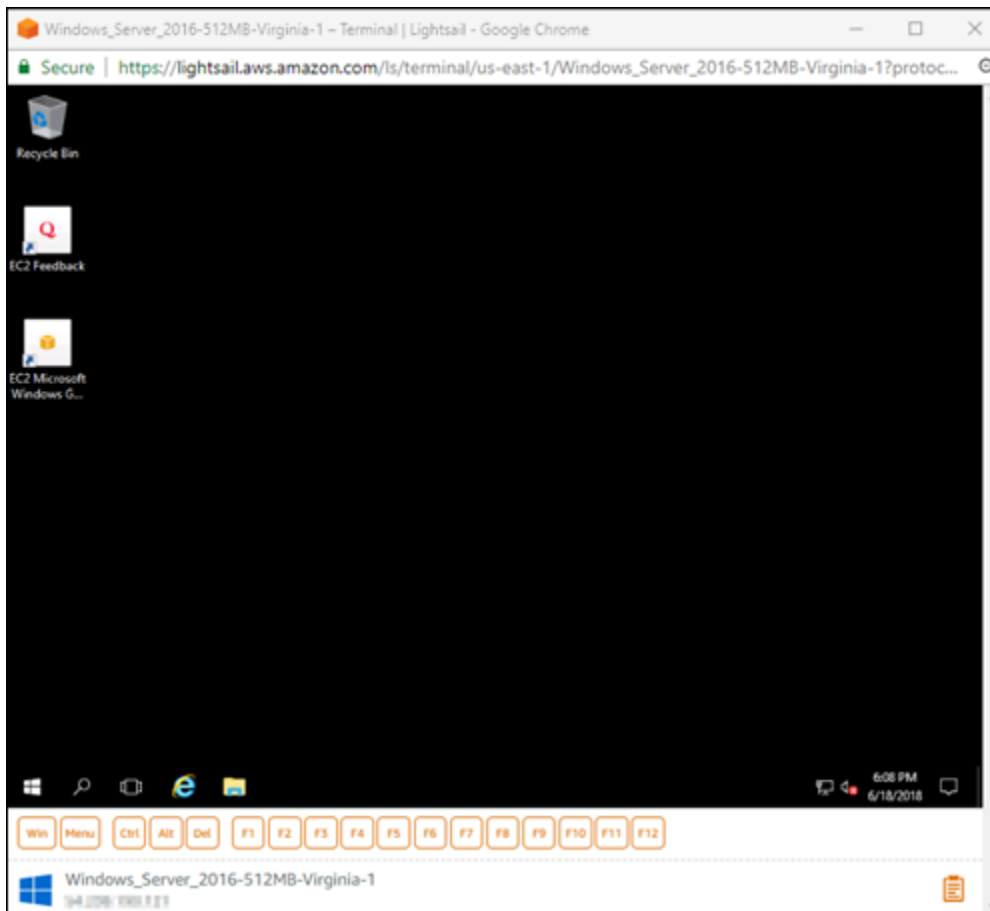
- Choose the actions menu icon (:), then choose **Connect** as shown in the following example.



- Choose the name of the instance, and on the **Connect** tab, choose **Connect using RDP**.



You can start interacting with your instance when the browser-based RDP client opens, and a Windows desktop is displayed as shown in the following example.



Note

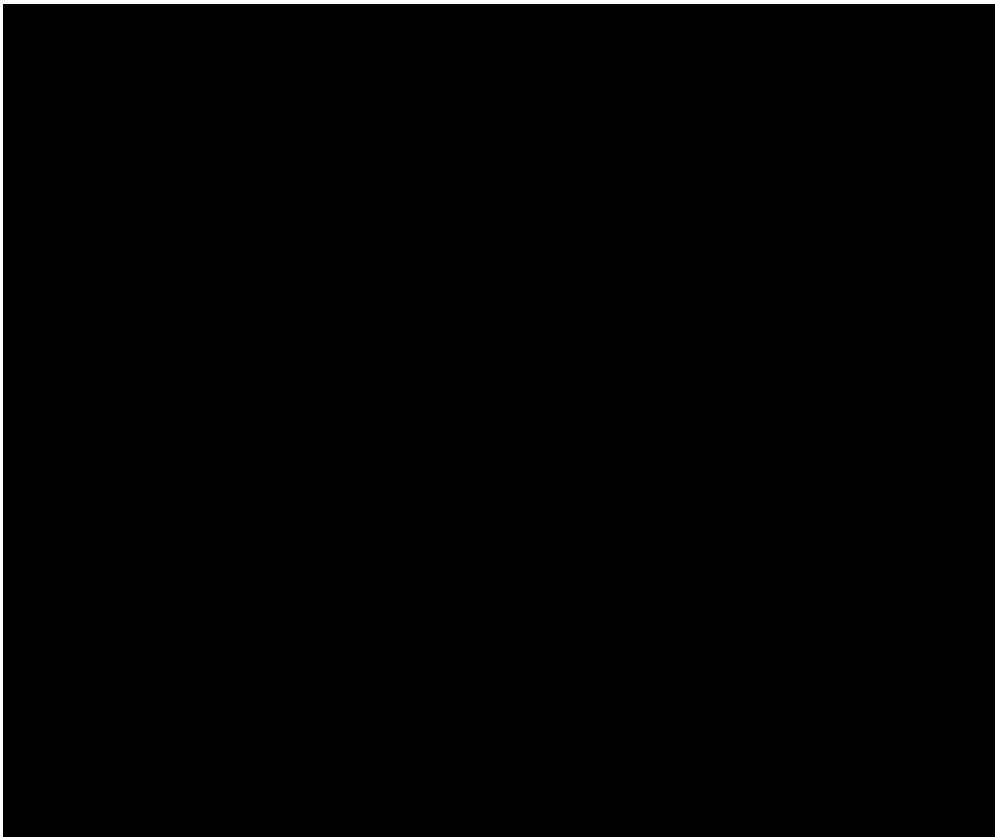
The **Connect** tab also provides the information required to connect using your own RDP client, such as the default user name and password for your Windows instance. For more information about configuring your own RDP client, see [Connecting to your Windows instance in Amazon Lightsail using the Remote Desktop Connection client](#).

Interact with your Windows instance using the browser-based RDP client

Use the browser-based RDP client as you would your own local Windows desktop. RDP includes function keys and other keys specific to Windows to help you interact with your instance. The following sections show you how to copy and paste text to and from the clipboard in RDP.

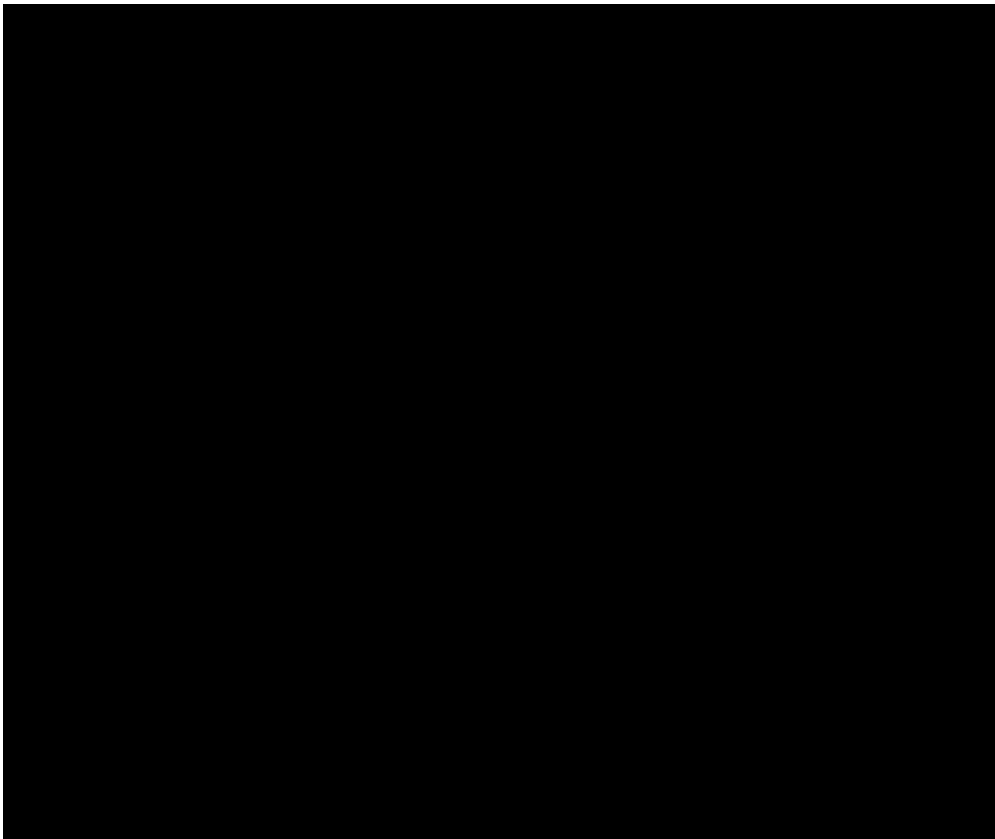
To paste text into the browser-based RDP client

1. Highlight text in your local desktop, then press **Ctrl+C** or **Cmd+C** to copy it to your local clipboard.
2. In the bottom right corner of the browser-based RDP client, choose the clipboard icon. The browser-based RDP client clipboard text box appears.
3. Click into the text box, then press **Ctrl+V** or **Cmd+V** to paste the contents from your local clipboard into the browser-based RDP client clipboard.
4. Right-click any area on the remote desktop screen to paste the text from the browser-based RDP client clipboard to the remote desktop screen.



To copy text from the browser-based RDP client

1. Highlight text on the remote desktop screen.
2. In the bottom right corner of the browser-based RDP client, choose the clipboard icon. The browser-based RDP client clipboard text box appears.
3. Highlight the text that you want to copy, then press **Ctrl+C** or **Cmd+C** to copy the text to your local clipboard. You can now paste the copied text anywhere in your local desktop.



Change the Administrator password for a Lightsail Windows instance

When you create a Windows Server-based Lightsail instance, we use the default password for the AWS Region where we create the instance. This makes it easier to connect using the browser-based remote desktop (RDP) client, as well as a client such as Remote Desktop Connection.

Important

We strongly encourage you to let Lightsail generate the password for your instance. Since we don't store your custom password, you can risk losing access to your Lightsail instance if you change the Administrator password.

Change your Administrator password using Windows Server

You can change your Administrator password using the Windows Server **Change Password** tool. Type **Ctrl + Alt + Del** on your Windows Server-based Lightsail instance, and then choose **Change a password**.

Decrypt your key

If you change your password on your Windows Server-based Lightsail instance, you can use the AWS Command Line Interface (AWS CLI) to get information that helps you decrypt your password.

Get your ciphertext using the AWS CLI

1. If you haven't done so already, install and configure the AWS CLI.

For more information, see [Configure the AWS Command Line Interface to work with Amazon Lightsail](#).

2. Open a command prompt or a terminal.
3. Type the following command.

```
aws lightsail get-instance-access-details --instance-name my-instance
```

Where *my-instance* is the name of the instance you want to get information about.

You'll see output similar to the following.

```
{
  "accessDetails": {
    "username": "Administrator",
    "protocol": "rdp",
    "ipAddress": "12.345.678.910",
    "passwordData": {
      "ciphertext": "cipher",
      "keyPairName": "my-ohio-key"
    },
    "password": "",
    "instanceName": "2016-ohio-windows"
  }
}
```

4. You can use the ciphertext with any available application to decrypt your password.

Connect to a Lightsail Windows instance from Windows using Remote Desktop Connection

You can use the Remote Desktop Connection (RDC) client included with the Windows operating system to connect to your Windows instance in Amazon Lightsail. RDC requires that you use the administrator user name and password for the Windows instance, which could be the default password assigned to the instance when it's created or your own password if you changed the default password.

This topic walks you through the steps to obtain your default administrator password from the Lightsail console, and configure RDC to connect to your Windows instance. You can also connect to your instance from within the Lightsail console using your browser. For more information, see [Connect to your Windows instance with the web-based RDP client](#).

Get the default administrator password for your Windows instance

Complete the following steps to get the default administrator password for your Windows instance, which is required to connect to the instance using RDC.

Note

If you changed the default administrator password, then the password that is displayed in Lightsail console for your instance will not work. You'll need to remember your password. You cannot connect to your instance using RDC without your administrator password.

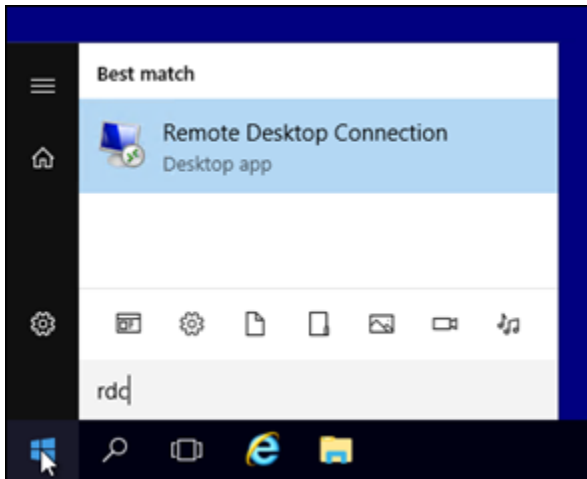
1. Sign in to the [Lightsail console](#).
2. Choose the Windows instance that you want to connect to.
3. In the **Connect** tab of the instance management page, choose **Show default password**.
4. Highlight the default password that is displayed, and copy it by pressing **Ctrl+Cor Cmd+C**. The password is now in your clipboard.

Continue to the next section of this guide to configure RDC, and paste the password into the client.

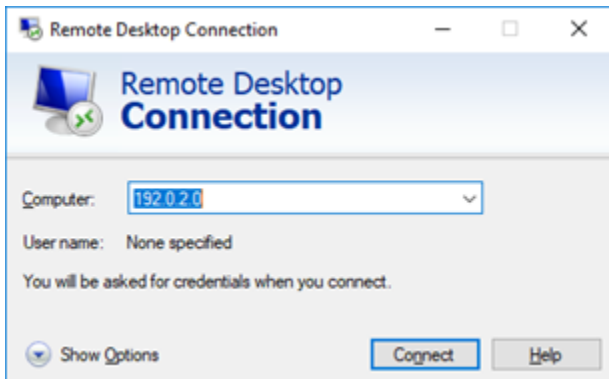
Configure RDC and connect to your Windows instance

Complete the following steps to configure RDC and connect to your Windows instance.

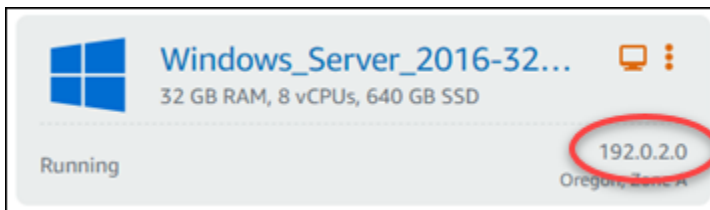
1. Open the Windows menu, and then search for Remote Desktop Connection or RDC.
2. Choose **Remote Desktop Connection** in the search results.



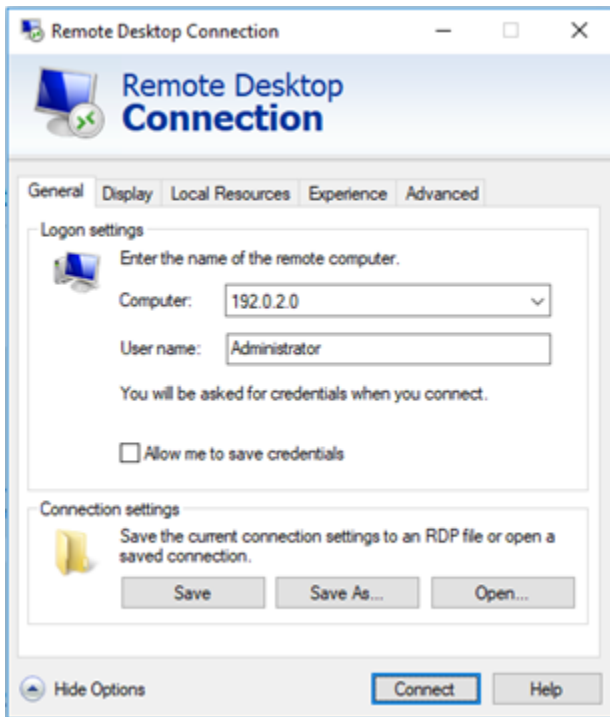
3. In the **Computer** text box, enter your Windows instance's public IP address.



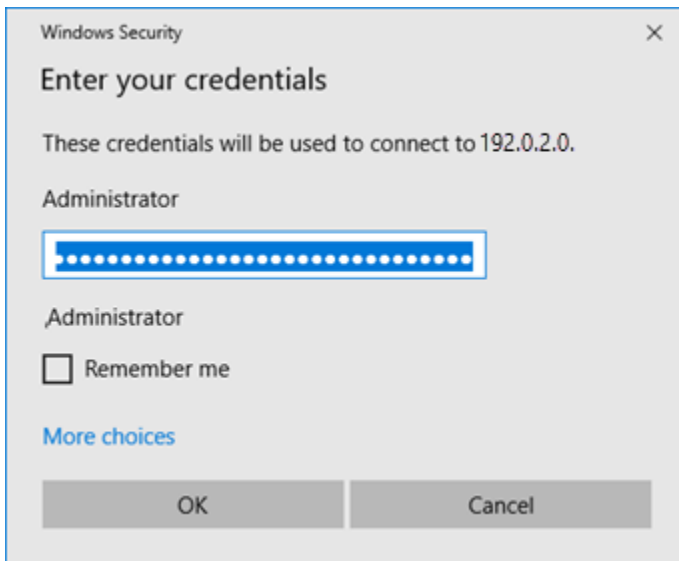
The public IP is displayed next to your instance in the Lightsail console, as shown in the following example:



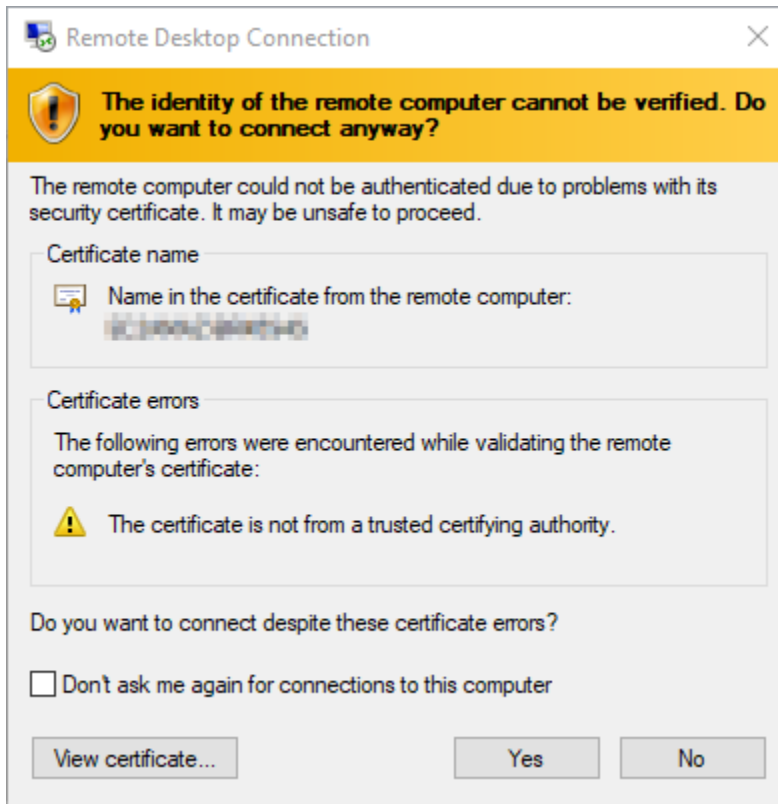
4. Choose **Show Options** to view additional connection options.
5. In the **User Name** text box, enter Administrator, which is the default user name for all Windows instances in Lightsail.



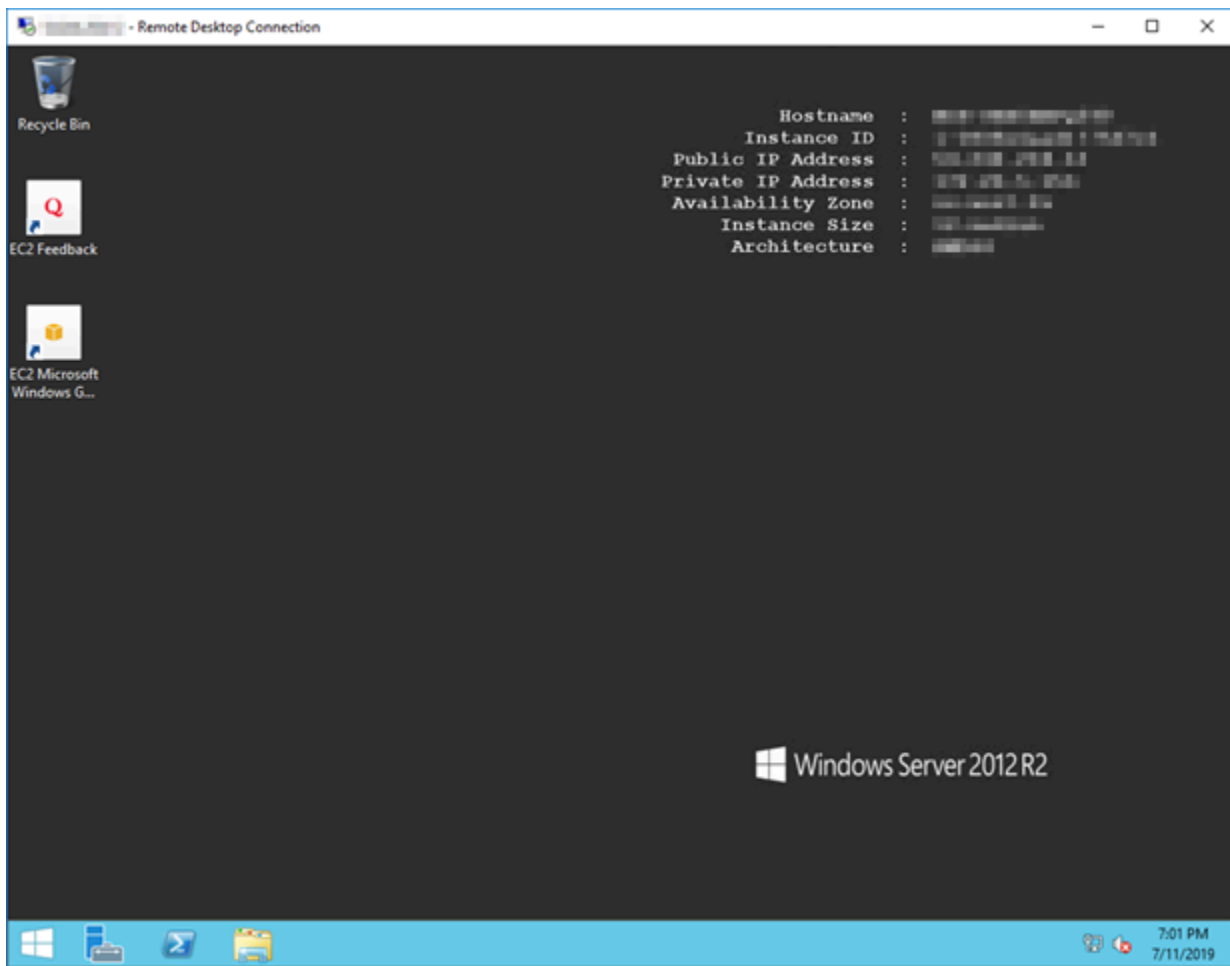
6. Choose **Connect**.
7. In the prompt that appears, enter or paste the default administrator password that you copied from the Lightsail console earlier in this procedure, and then choose **OK**.



8. In the prompt that appears, choose **Yes** to connect to the Windows instance despite certificate errors.



After you're connected to the instance, you should see a screen similar to the following example:



Connect to a Lightsail Windows instance from macOS using Remote Desktop Connection

You can use the Microsoft Remote Desktop client to connect to your Windows instance from your macOS computer. Microsoft Remote Desktop requires that you use the administrator user name and password for your Lightsail Windows instance. This can be the default password assigned to the instance when it is created, or your own password if you changed the default password.

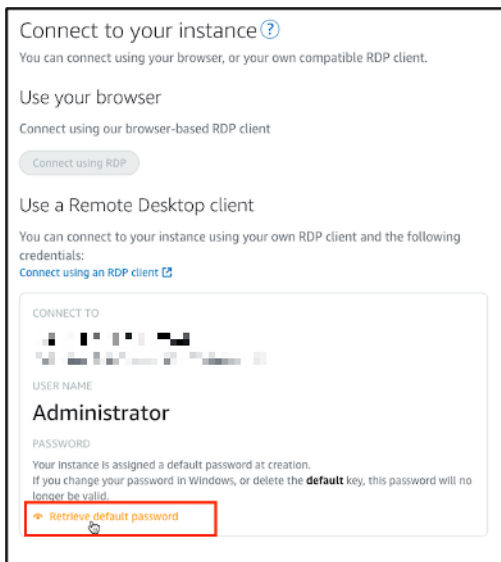
This topic walks you through the steps to obtain your default administrator password from the Lightsail console, and configure Microsoft Remote Desktop to connect to your Windows instance. You can also connect to your instance from within the Lightsail console using your browser. For more information, see [Connect to your Windows instance with the Microsoft Remote Desktop client](#).

Get the required connection information for your Windows instance

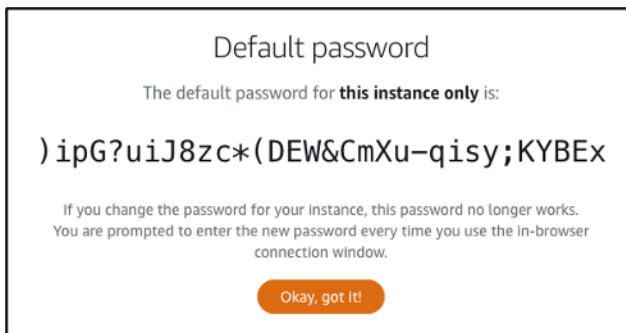
You will need the public IP address, user name, and administrator password for your Windows instance to connect to it using the Microsoft Remote Desktop client.

Complete the following procedure to get the required information.

1. Sign in to the [Lightsail console](#).
2. Choose the **Instances** tab on the Lightsail home page.
3. Make note of the public IP address of the instance you want to connect to.
4. Choose the name of the instance you want to connect to.
5. Choose the **Connect** tab.
6. Choose **Show default password** to obtain the Windows administrator password for your instance.



The prompt displays the default administrator password for your Windows instance.

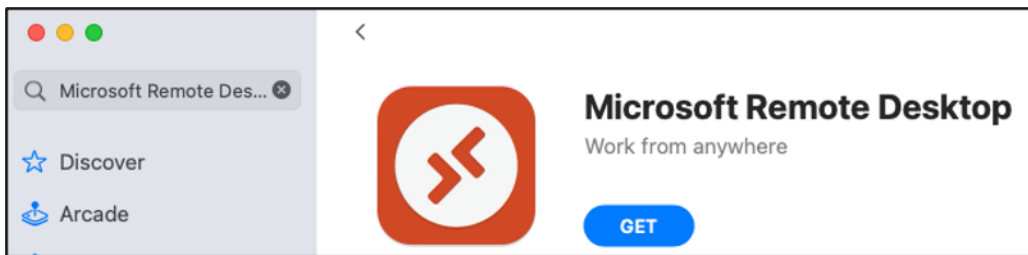


7. Copy the administrator password. You will use it to sign in to your instance using the Microsoft Remote Desktop client later in this guide.

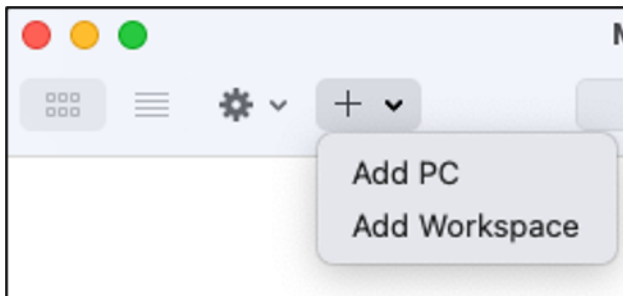
Configure Microsoft Remote Desktop and connect to your instance

Complete the following procedure to install the Microsoft Remote Desktop client on your Mac, and configure it to connect to your instance.

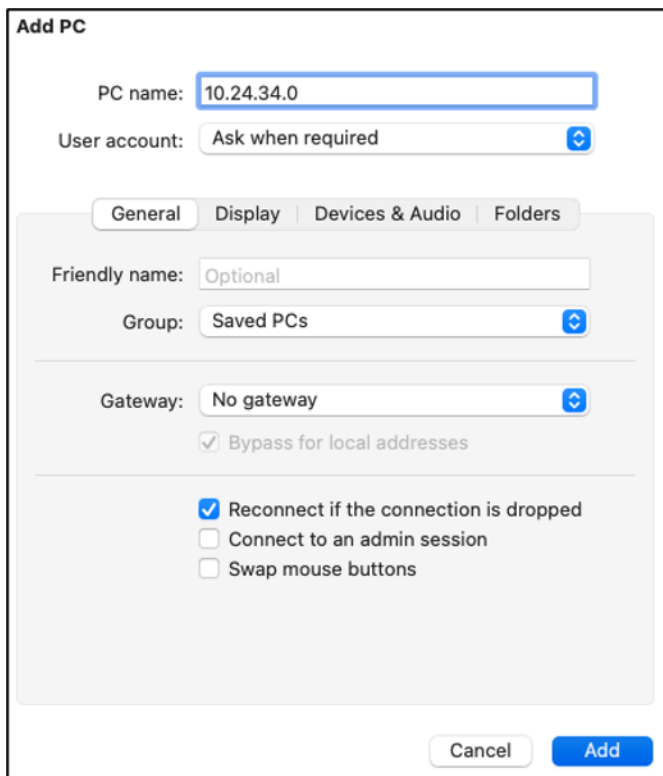
1. Open the App Store on your Mac, and search for **Microsoft Remote Desktop**.
2. Find the **Microsoft Remote Desktop** app in the search results, and choose **GET** to install the application.



3. Open **Microsoft Remote Desktop** after the installation is complete.
4. At the top, choose the **plus (+)** icon, and choose **Add PC**.



5. In the **PC name** text box, paste the public IP address of your instance.
6. Choose **Add**.



Add PC

PC name: 10.24.34.0

User account: Ask when required

General | Display | Devices & Audio | Folders

Friendly name: Optional

Group: Saved PCs

Gateway: No gateway

Bypass for local addresses

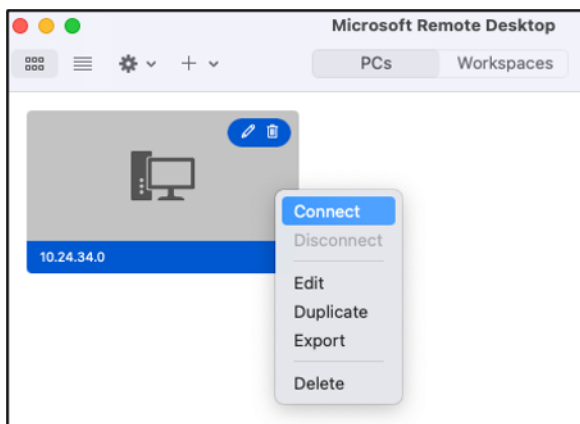
Reconnect if the connection is dropped

Connect to an admin session

Swap mouse buttons

Cancel Add

7. Right-click the icon for your instance, and choose **Connect**.



8. Enter **Administrator** into the **Username** text box, and enter the default administrator password that you got earlier in this guide into the **Password** text box.
9. Choose **Continue** to connect to your instance.

Enter Your User Account

This user account will be used to connect to 204.236.212.128 (remote PC).

Username:

Password:

Show password

You are now connected to your Lightsail Windows instance.



Create a snapshot of your Linux or Unix Lightsail instance

You can create snapshots of your Linux/Unix-based Lightsail instances. An *instance snapshot* is a copy of the system disk and matches the original machine configuration (memory, CPU, disk size, and data transfer rate). If you've attached block storage disks to your instance, Lightsail copies those additional disks as part of your snapshot. For more information, see [Snapshots](#).

Note

The steps to create a snapshot of a Windows Server-based Lightsail instance are different. For more information, see [Create a snapshot of your Windows Server instance](#).

You must already have an instance in Lightsail to create a snapshot of it. After you have an instance, follow these steps to create a snapshot:

1. On the Lightsail home page, choose the name of your instance for which you want to create a snapshot.
2. Choose the **Snapshots** tab.
3. Under the **Manual snapshots** section of the page, choose **Create snapshot**, then enter a name for your snapshot.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
4. Choose **Create**.

You can see the snapshot you just created with a status of **Snapshotting....**

After the snapshot is finished, you can [create another instance from the snapshot](#). For example, you may want to choose a larger size bundle than you had previously.

Important

When you create a new instance from a snapshot, Lightsail lets you create an instance bundle that is either the same size or larger size. We do not currently support creating a *smaller* instance size from a snapshot. The smaller options will be grayed out when you create a new instance from a snapshot.

To create a larger instance size from a snapshot, you can use the Lightsail console, the **create-instances-from-snapshot** CLI command, or the **CreateInstancesFromSnapshot** API operation. For more information, see [Create an instance from a snapshot](#).

For more information about Lightsail bundles, see [Lightsail pricing](#).

Topics

- [Connect to a Linux or Unix instance in Amazon EC2 created from an Amazon Lightsail snapshot](#)
- [Connect to a Windows Server instance in Amazon EC2 created from a Lightsail snapshot](#)
- [Create a snapshot of your Lightsail Windows Server instance](#)

- [Secure a Windows Server instance in Amazon EC2 created from a Lightsail snapshot](#)
- [Secure a Linux or Unix instance in Amazon EC2 created from a Lightsail snapshot](#)

Connect to a Linux or Unix instance in Amazon EC2 created from an Amazon Lightsail snapshot

After a Linux or Unix instance is created in Amazon Elastic Compute Cloud (Amazon EC2) from an Amazon Lightsail snapshot, you can connect to the instance via SSH similar to how you connected to the source Lightsail instance. To authenticate to your instance, use either the default Lightsail key pair for the source instance's AWS Region, or your own key pair. This guide shows you how to connect to your Linux or Unix instance in EC2 using PuTTY.

Note

For more information about connecting to a Windows Server instance, see [Connect to an Amazon EC2 Windows Server instance that was created from a Lightsail snapshot](#).

Contents

- [Get the key for your instance](#)
- [Get the public DNS address for your instance](#)
- [Download and install PuTTY](#)
- [Configure the key with PuTTYgen](#)
- [Configure PuTTY to connect to your instance](#)
- [Next steps](#)

Get the key for your instance

Get the correct key required to connect to your new Amazon EC2 instance. The key that you need depends on how you connected to the source Lightsail instance. You may have connected to the source Lightsail instance using one of the following methods:

- **Using the default Lightsail key pair for the source instance's Region** — Download the default private key from the **SSH keys** tab on the [Lightsail account page](#). For more information about the default Lightsail keys, see [SSH key pairs](#).

Note

After you connect to your EC2 instance, we recommend removing the default Lightsail key from the instance and replacing it with your own key pair. For more information, see [Secure your Linux or Unix instance in Amazon EC2 created from a Lightsail snapshot](#).

- **Using your own key pair** — Locate your private key and use it to connect to your Amazon EC2 instance. Lightsail does not store your private key when you use your own key pair. If you've lost your private key, you cannot connect to your Amazon EC2 instance.

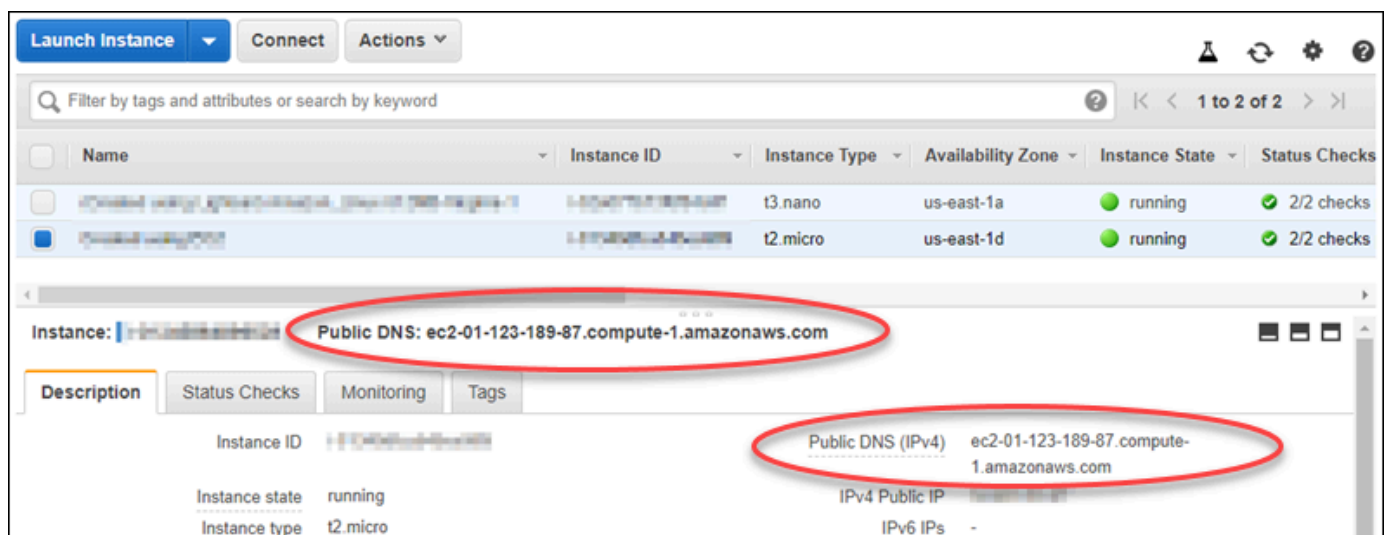
Get the public DNS address for your instance

Get the public DNS address for your Amazon EC2 instance, so that you can use it when configuring an SSH client, such as PuTTY, to connect to your instance.

To get the public DNS address for your instance

1. Sign in to the [Amazon EC2 console](#).
2. Choose **Instances** from the left navigation pane.
3. Choose the running Linux or Unix instance that you want to connect to.
4. In the lower pane, locate the **Public DNS** address for your instance.

This is the address that you will use when configuring an SSH client to connect to your instance. Continue to the [Download and install PuTTY](#) section of this guide to learn how to download and install the PuTTY SSH client.



Download and install PuTTY

PuTTY is a free SSH client for Windows. For more information about [PuTTY, see PuTTY: a free SSH and Telnet client](#). This website also describes the restrictions in countries where encryption isn't allowed. If you already have PuTTY, you can skip to the following *Configure the key with PuTTYgen* section of this guide.

[Download the PuTTY installer or executable file](#). We recommend using the latest version. However, for information about which download to choose, see the [PuTTY documentation](#).

Continue to the [Configure the key with PuTTYgen](#) section of this guide to configure the key with PuTTYgen.

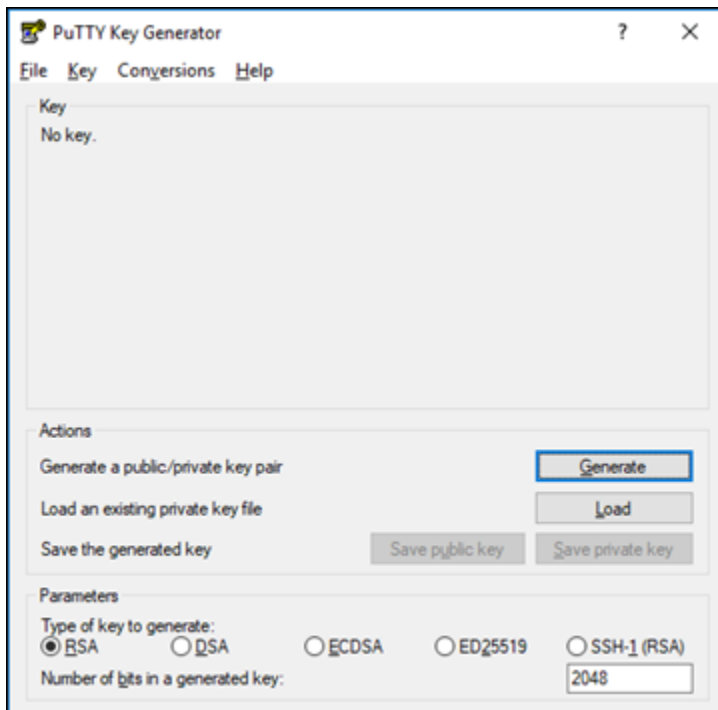
Configure the key with PuTTYgen

PuTTYgen generates pairs of public and private keys to be used with PuTTY. This step is required to use the key file type (.PPK) that PuTTY accepts.

To configure the key with PuTTYgen

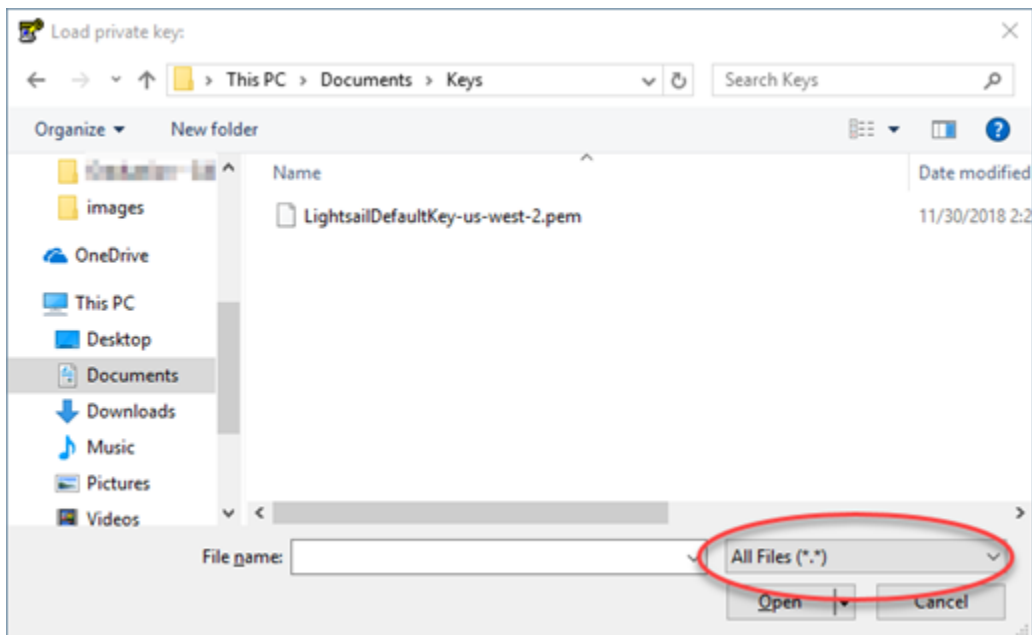
1. Start PuTTYgen.

For example, choose the **Windows Start** menu, choose **All Programs**, choose **PuTTY**, and choose **PuTTYgen**.

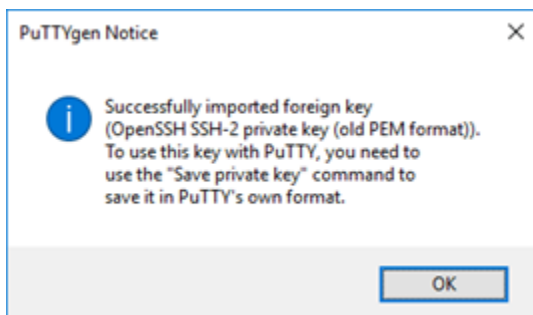


2. Choose **Load**.

By default, PuTTYgen displays only files with the .PPK extension. To locate your .PEM file, select the option to display files of all types.

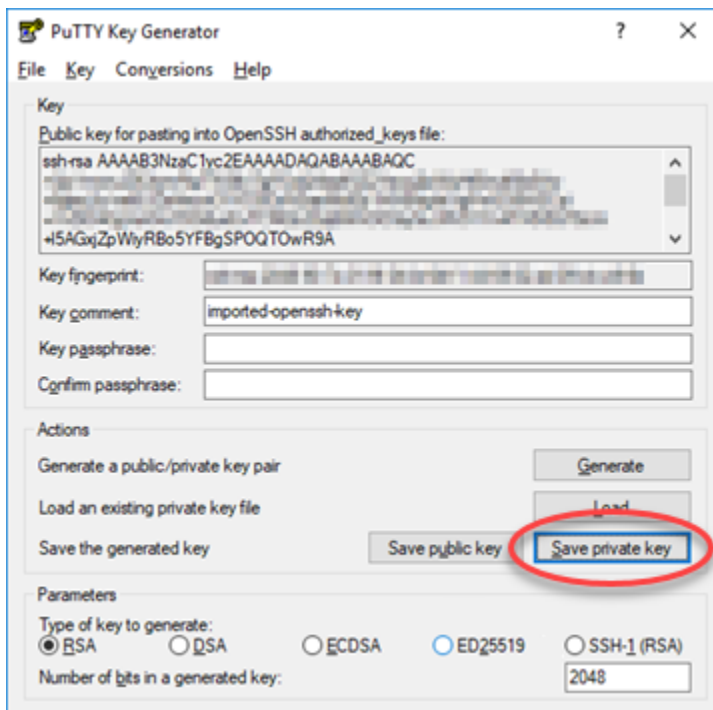


3. Choose the default Lightsail key file (.PEM) that you downloaded earlier in this guide, and then choose **Open**.
4. After PuTTYgen confirms that you successfully imported the key, choose **OK**.



5. Choose **Save private key**, and then confirm that you don't want to save it with a passphrase.

If you create a passphrase as an extra measure of security, you must enter it every time you connect to your instance using PuTTY.



6. Specify a name and a location to save your private key, and then choose **Save**.

PuTTYgen saves your new key file as a .PPK file type.

7. Close PuTTYgen.

Continue to the [Configure PuTTY to connect to your instance](#) section of this guide to use the new .PPK file that you generated to configure PuTTY and connect to your Linux or Unix instance in Amazon EC2.

Configure PuTTY to connect to your instance

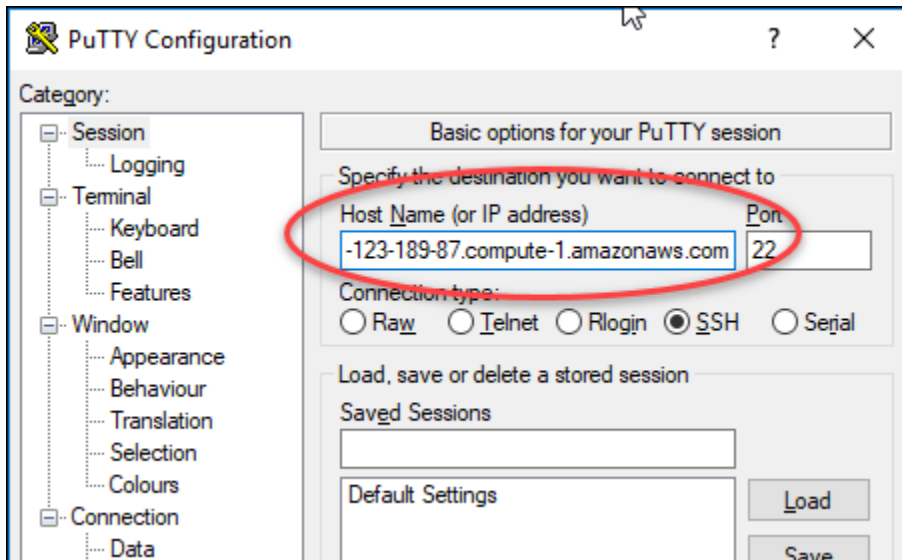
Configure PuTTY, now that you have all of the requirements to connect to your Linux or Unix instance using SSH.

To configure PuTTY to connect to your Linux or Unix instance

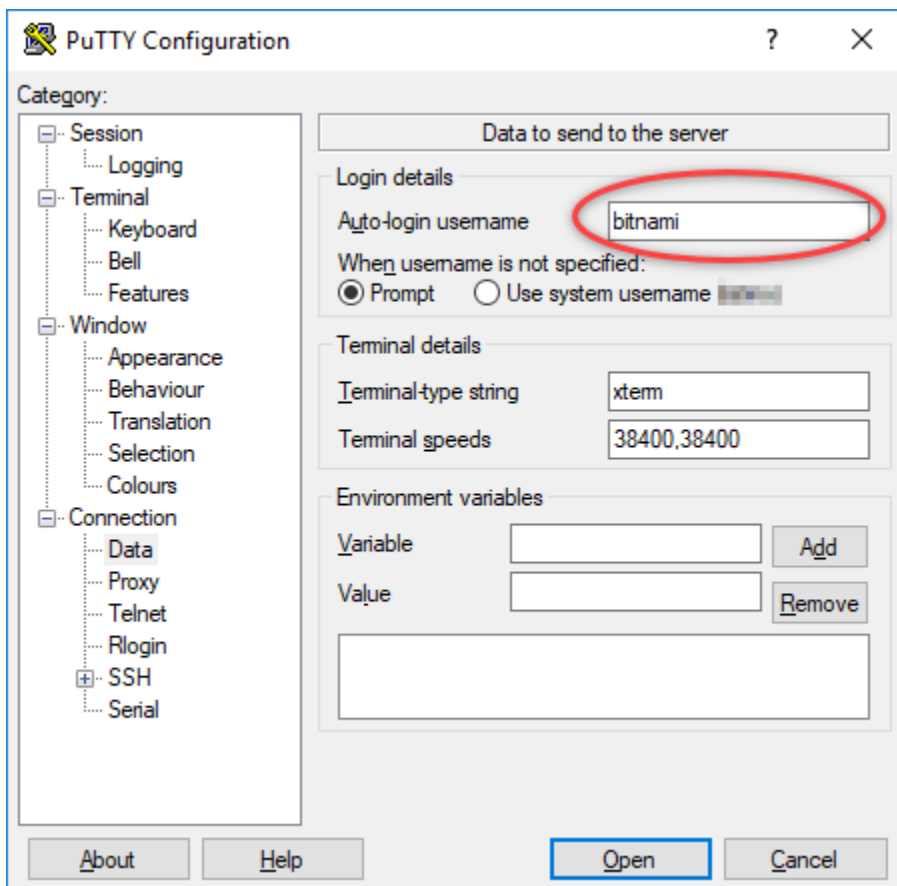
1. Open PuTTY.

For example, choose the **Windows Start** menu, choose **All Programs**, choose **PuTTY**, and choose **PuTTY**.

2. In the **Host Name** text box, enter the public DNS address for your instance that you obtained from the Amazon EC2 console earlier in this guide.

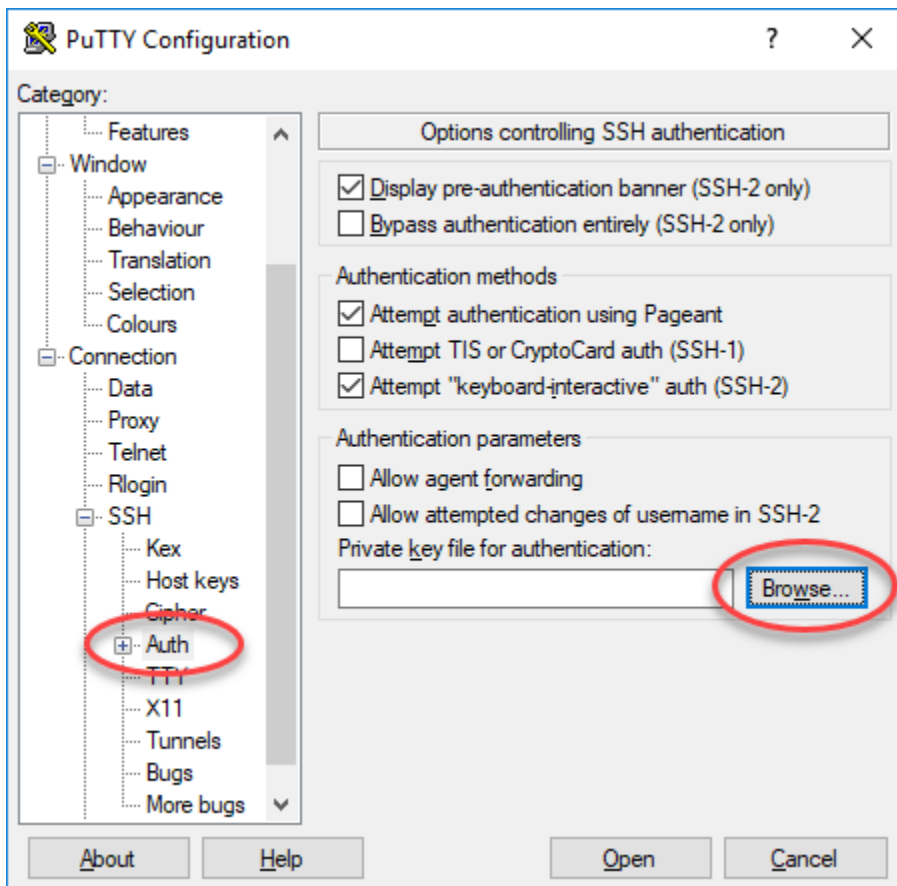


3. Under the **Connection** section in the left navigation pane, choose **Data**.
4. In the **Auto-login username** text box, enter a user name to use when logging in to the instance.



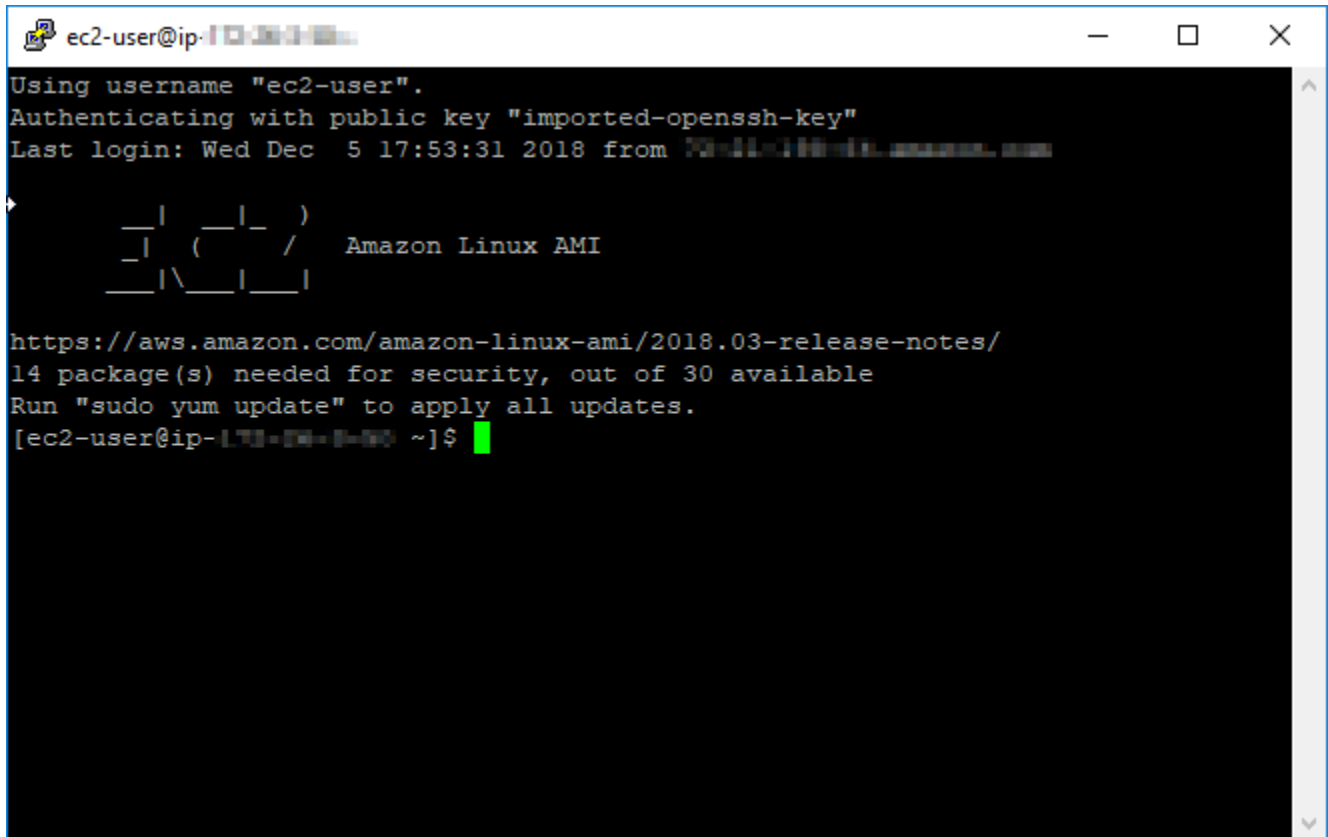
Enter one of the following default user names depending on the blueprint of the source Lightsail instance:

- AlmaLinux, Amazon Linux 2, Amazon Linux 2023, CentOS Stream 9, FreeBSD, and openSUSE instances: `ec2-user`
 - CentOS 7 instances: `centos`
 - Debian instances: `admin`
 - Ubuntu instances: `ubuntu`
 - Bitnami instances: `bitnami`
 - Plesk instances: `ubuntu`
 - cPanel & WHM instances: `centos`
5. Under the **Connection** section in the left navigation pane, expand **SSH**, and then choose **Auth**.
 6. Choose **Browse** to navigate to the .PPK file that you created in the previous section of this guide, and then choose **Open**.



7. Choose **Open** to connect to your instance, and then choose **Yes** to trust this connection in the future.

You should see a screen similar to the following if you've successfully connected to your instance:



```
ec2-user@ip-173-20-1-90:~$ ssh -i /home/ec2-user/.ssh/imported-openssh-key ec2-user@ip-173-20-1-90
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Wed Dec  5 17:53:31 2018 from 72.21.218.100:22: ssh://ec2-user@ip-173-20-1-90

  _ | _ | _ )
  _ | ( _ | /   Amazon Linux AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
14 package(s) needed for security, out of 30 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-173-20-1-90 ~]$
```

Next steps

Your new Linux or Unix instance in Amazon EC2 contains residual keys from the Lightsail service, if you use Amazon EC2 to create new instances from your exported snapshots. We recommend removing these keys to enhance security for your new Amazon EC2 instance. For more information, see [Secure your Linux or Unix instance in Amazon EC2 created from a Lightsail snapshot](#).

Connect to a Windows Server instance in Amazon EC2 created from a Lightsail snapshot

After your new Windows Server instance is created in Amazon Elastic Compute Cloud (Amazon EC2), you can connect to it using Remote Desktop Protocol (RDP). This is similar to how you connected to the source Amazon Lightsail instance. Connect to your EC2 instance using the default

Lightsail key pair for the source instance's AWS Region. This guide shows you how to connect to your Windows Server instance using Microsoft Remote Desktop Connection.

Note

For more information about connecting to a Linux or Unix instance, see [Connect to a Linux or Unix instance in Amazon EC2 created from a Lightsail snapshot](#).

Contents

- [Get the key for your instance](#)
- [Get the public DNS address for your instance](#)
- [Get the password for your Windows Server instance](#)
- [Configure Remote Desktop Connection to connect to your Windows Server instance](#)
- [Next steps](#)

Get the key for your instance

Your Windows Server instance in Amazon EC2 uses the default Lightsail key pair for the source instance's Region to retrieve the default administrator password.

Download the default private key from the **SSH keys** tab on the [Lightsail account page](#). For more information about the default Lightsail SSH keys, see [SSH key pairs](#).

Note

After you connect to your EC2 instance, we recommend changing the administrator password for your Windows Server instance in Amazon EC2. It removes the association between the default Lightsail key pair and your Windows Server instance in Amazon EC2. For more information, see [Secure an Amazon EC2 Windows Server instance that was created from a Lightsail snapshot](#).

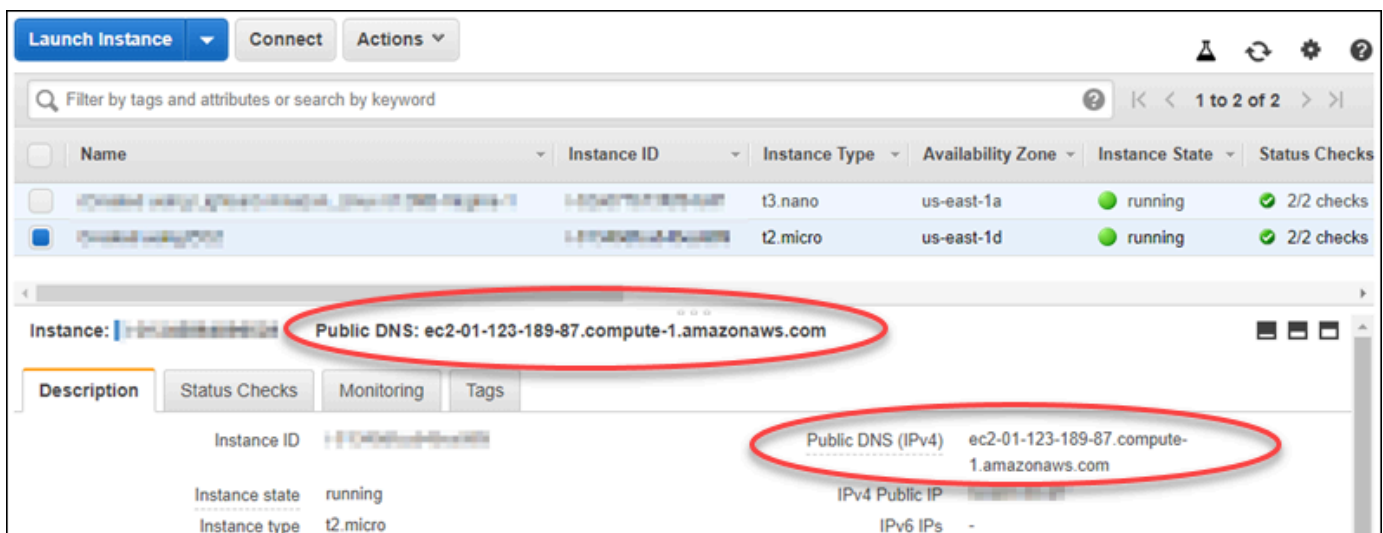
Get the public DNS address for your instance

Get the public DNS address for your Amazon EC2 instance, so that you can use it when configuring an RDP client, such as Microsoft Remote Desktop Connection, to connect to your instance.

To get the public DNS address for your instance

1. Sign in to the [Amazon EC2 console](#).
2. Choose **Instances** from the left navigation pane.
3. Choose the running Windows Server instance that you want to connect to.
4. In the lower pane, locate the **Public DNS** address for your instance.

This is the address that you use when configuring an RDP client to connect to your instance. Continue to the [Get the password for your Windows Server instance](#) section of this guide to learn how to get the default administrator password for your Windows Server instance in Amazon EC2.

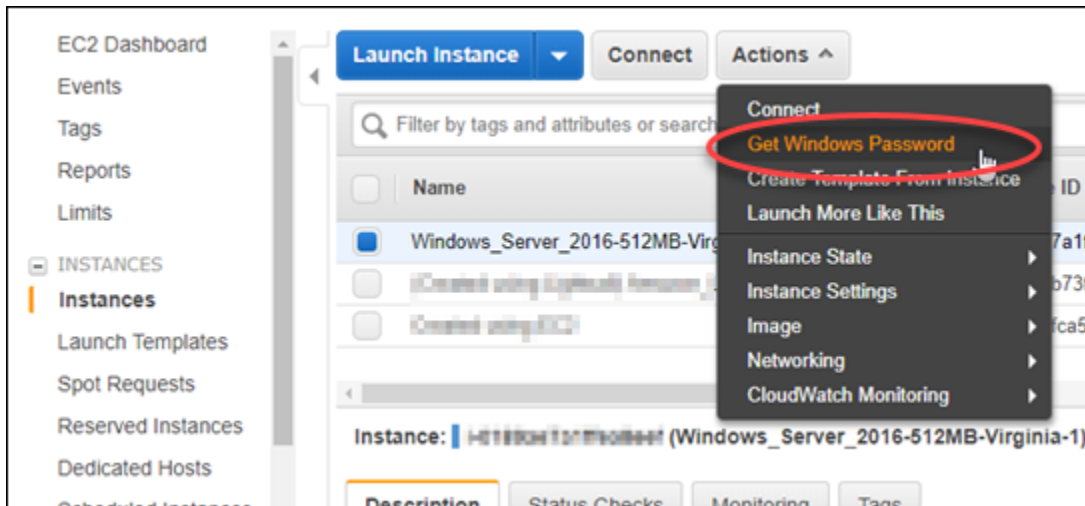


Get the password for your Windows Server instance

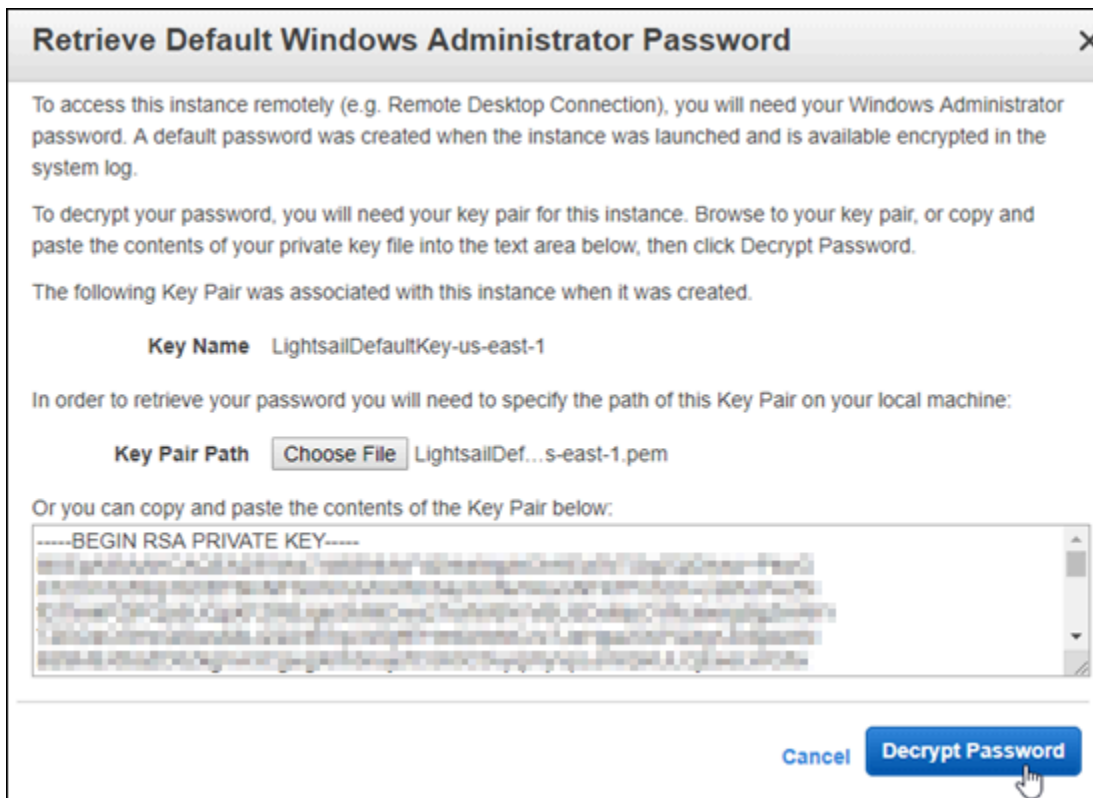
Get the password for your Windows Server instance from the Amazon EC2 console. You need this password to sign in to your Windows Server instance when connecting to it through RDP.

To get the password for your Windows Server instance

1. Sign in to the [Amazon EC2 console](#).
2. From the left navigation pane, choose **Instances**.
3. Choose the Windows Server instance that you want to connect to.
4. Choose **Actions**, then choose **Get Windows Password**.



5. At the prompt, choose **Browse** and open the default private key file that you downloaded from Lightsail earlier in this guide.
6. Choose **Decrypt Password**.



The password is displayed on the screen, as well as the public DNS and user name. Copy the password to your clipboard so that you can use it in the following [Configure Remote Desktop Connection to connect to your Windows Server instance](#) section of this guide. Highlight the password, and press **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using macOS.



Continue to the [Configure Remote Desktop Connection to connect to your Windows Server instance](#) section of this guide to learn how to configure Remote Desktop Connection to connect to your Windows Server instance in Amazon EC2.

Configure Remote Desktop Connection to connect to your Windows Server instance

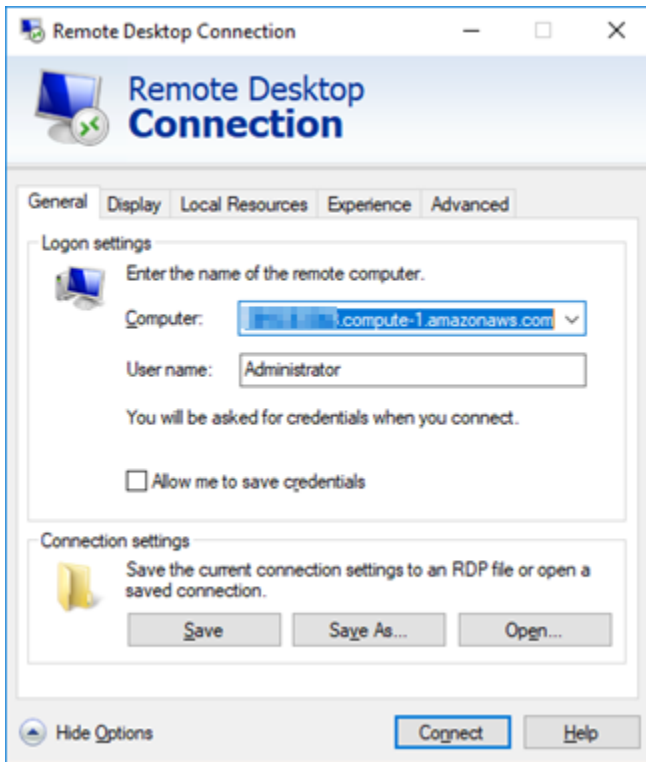
Remote Desktop Connection is an RDP client that comes pre-installed on most Windows operating systems. Use it to graphically connect to your Windows Server instance in Amazon EC2.

To configure Remote Desktop Connection to connect to your Windows Server instance

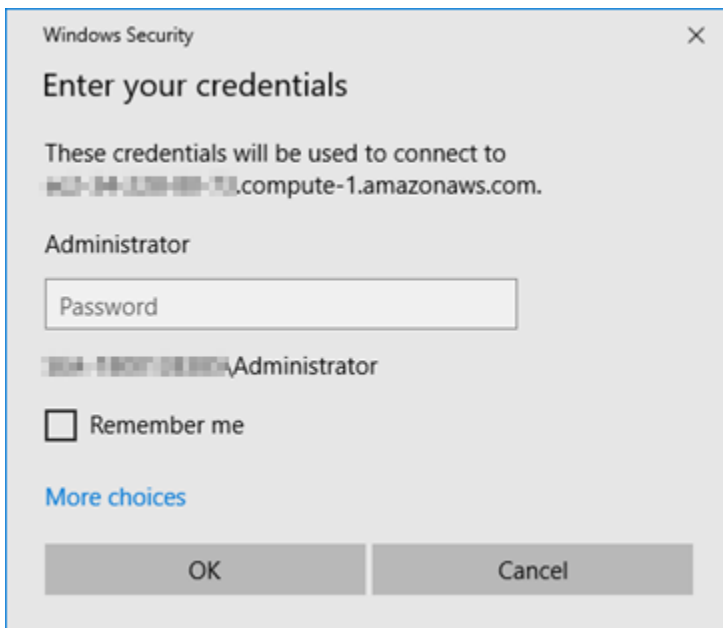
1. Open Remote Desktop Connection.

For example, choose the **Windows Start** menu, then search for **Remote Desktop Connection**.

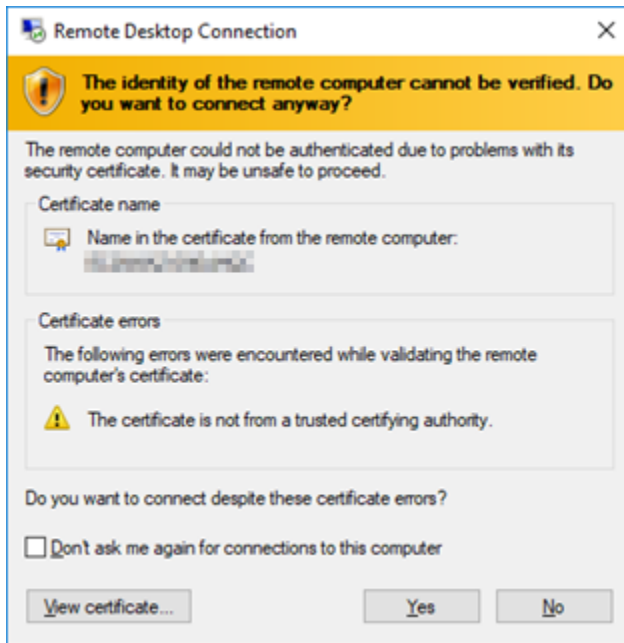
2. In the **Computer** text box, enter the public DNS address for your Windows Server instance in Amazon EC2 obtained earlier in this guide.
3. Choose **Show Options** to view additional options.
4. Enter Administrator into the **User name** text box.



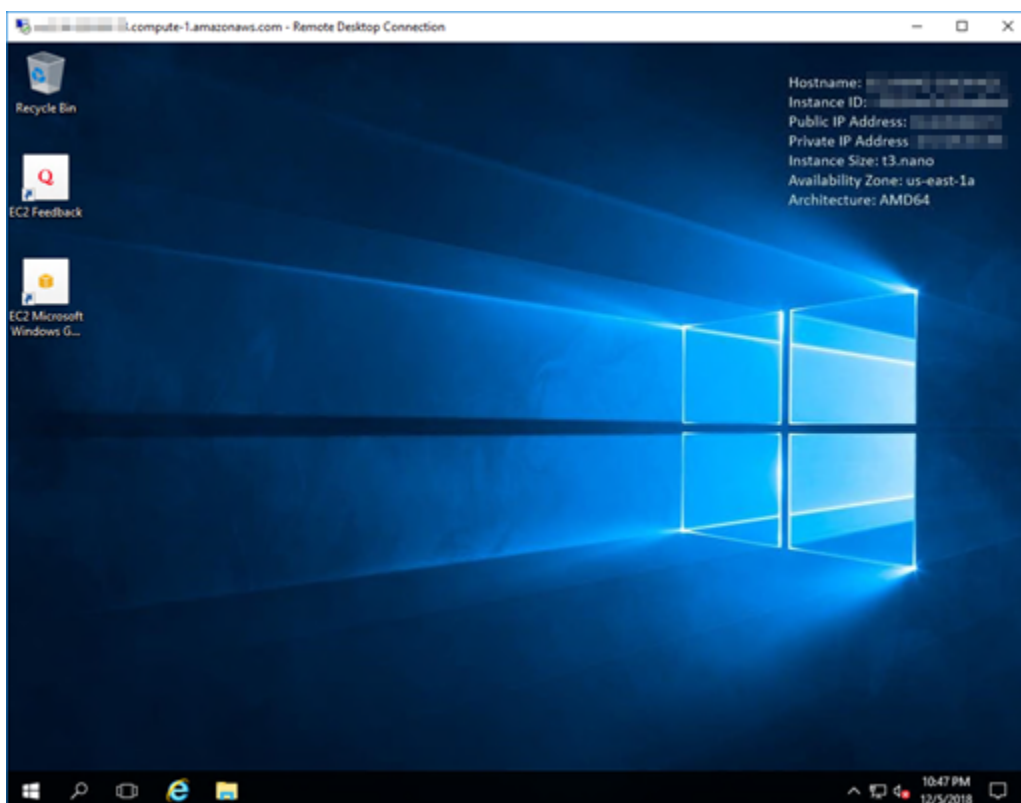
5. Choose **Connect** to connect to your Windows Server instance.
6. At the Windows Security prompt, enter the password for your Windows Server instance into the **Password** text box, then choose **OK**.



7. At the Remote Desktop Connection prompt, chose **Yes** to connect.



You should see a screen similar to the following if you've successfully connected to your instance:



Next steps

We recommend changing the administrator password for your Windows Server instance in Amazon EC2. It removes the association between the default Lightsail key pair and your Windows Server instance in Amazon EC2. For more information, see [Secure a Windows Server instance in Amazon EC2 created from a Lightsail snapshot](#).

Create a snapshot of your Lightsail Windows Server instance

A snapshot is a copy of the system disk and original configuration of an instance. The snapshot includes such information as memory, CPU, disk size, and data transfer rate. For more information, see [Snapshots](#).

To create a snapshot of your Windows Server instance in Lightsail, first create a backup snapshot. Next, create a second snapshot using a special utility known as System Preparation (Sysprep). Sysprep generalizes the Windows Server installation so that the instance can be backed up as a snapshot. Then, when you create an instance from that snapshot, you have an out-of-box experience as if you were running that Windows instance for the first time.

To create a snapshot of a Linux or Unix instance, see [Create a snapshot of your Linux or Unix instance](#).

Contents

- [Step 1: Create a backup snapshot before running Sysprep](#)
- [Step 2: Connect to your instance and shut it down using Sysprep](#)
- [Step 3: Create a snapshot after running Sysprep](#)

Step 1: Create a backup snapshot before running Sysprep

When you run Sysprep to create a snapshot, system-specific information is removed from your instance. This may have unintended consequences for the applications running on the instance. Therefore, you should first create a backup snapshot before running Sysprep to make sure that you have an alternate snapshot if something goes wrong.

When you create a snapshot before running Sysprep, instances that you create using the backup snapshot have the same administrator password as the original instance. You cannot connect to those instances using the browser-based RDP client in the Lightsail console. However, you can

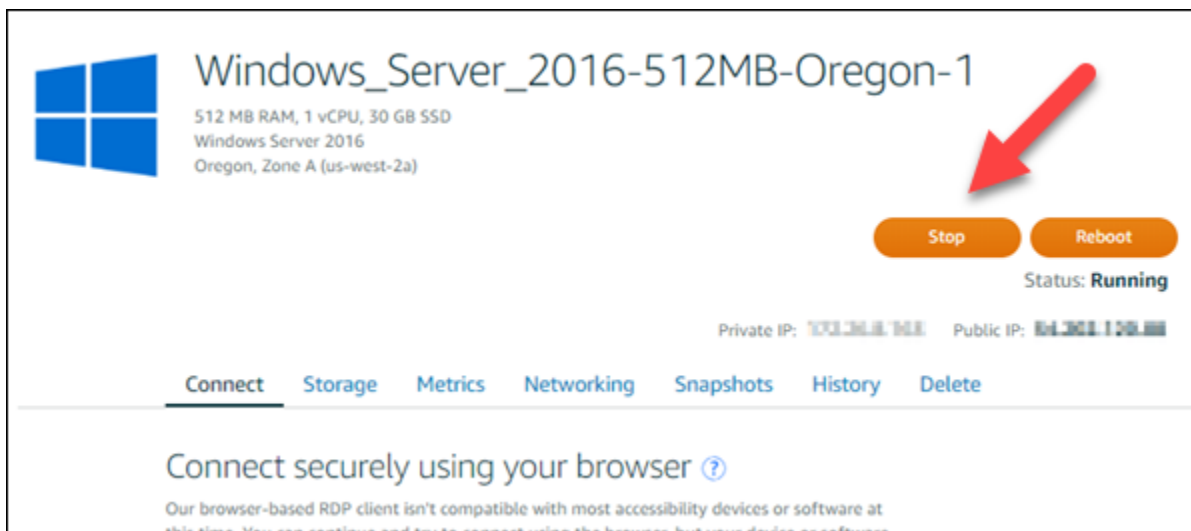
connect using your own RDP client and the same administrator password as the original instance. For more information, see [Connecting to your Windows instance in Amazon Lightsail using the Remote Desktop Connection client on a Windows computer](#).

Important

Save the administrator password of the original Windows instance and store it in a safe place. You will need that administrator password later if something goes wrong, and you create an instance from the snapshot you created before running Sysprep.

To create a backup snapshot before running Sysprep

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the name of the Windows Server instance for which you want to create a snapshot.
3. Choose **Stop** at the top of the instance management page to stop your instance.



Note

Stopping an instance makes any website or service on it unavailable until you start it again.

4. Choose the **Snapshots** tab.
5. Under the **Manual snapshots** section of the page, choose **Create snapshot**, then enter a name for your snapshot.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

6. Choose **Create**.

7. At the prompt, choose **Create snapshot** again to confirm.

The snapshot process takes a few minutes to complete.

8. After the snapshot is created, choose **Start** at the top of the instance management page to start your instance again.

Step 2: Connect to your instance and shut it down using Sysprep

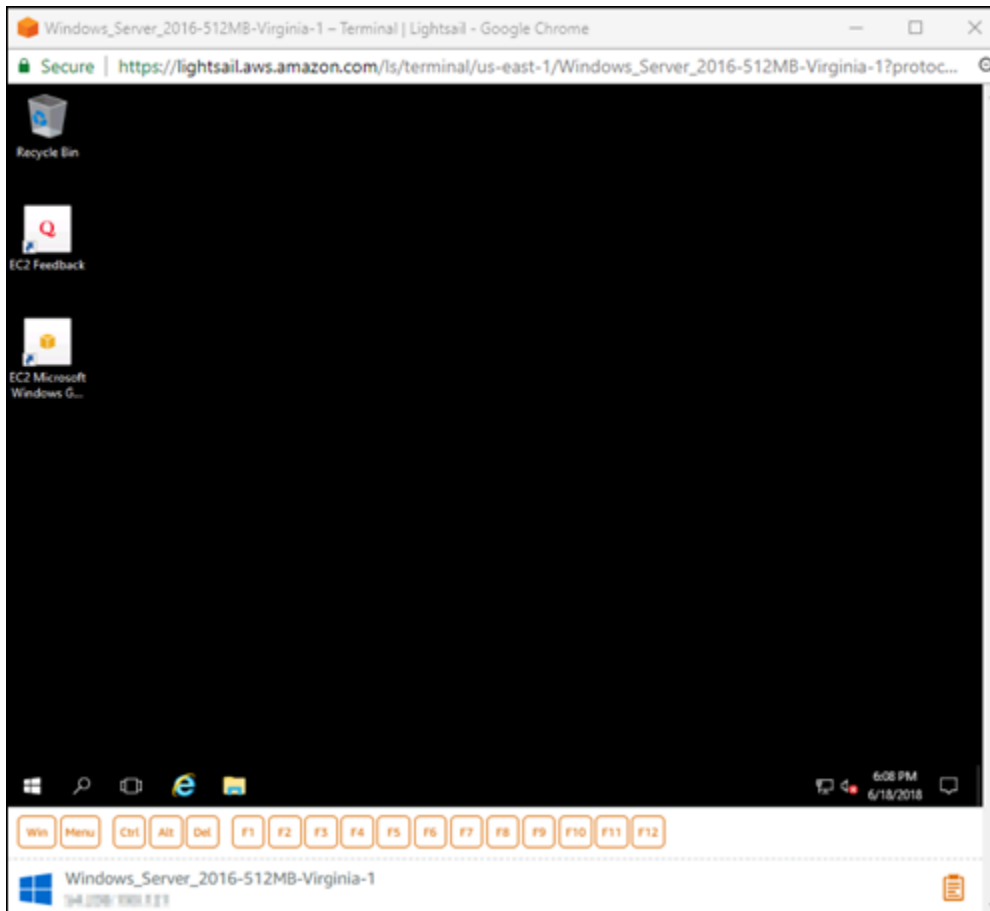
Now that you have a backup snapshot, it's time to run Sysprep on your Windows Server instance. This causes the instance to shut down so that you can take a snapshot. For more information about Sysprep, see [Sysprep Overview](#) in the Microsoft documentation.

In this step, connect to your instance and run Sysprep through a preinstalled application. The application is called **EC2LaunchSettings** on Windows Server 2019 and Windows Server 2016 instances, and **Ec2ConfigService Settings** on Windows Server 2012 instances.

To connect to your instance and run Sysprep

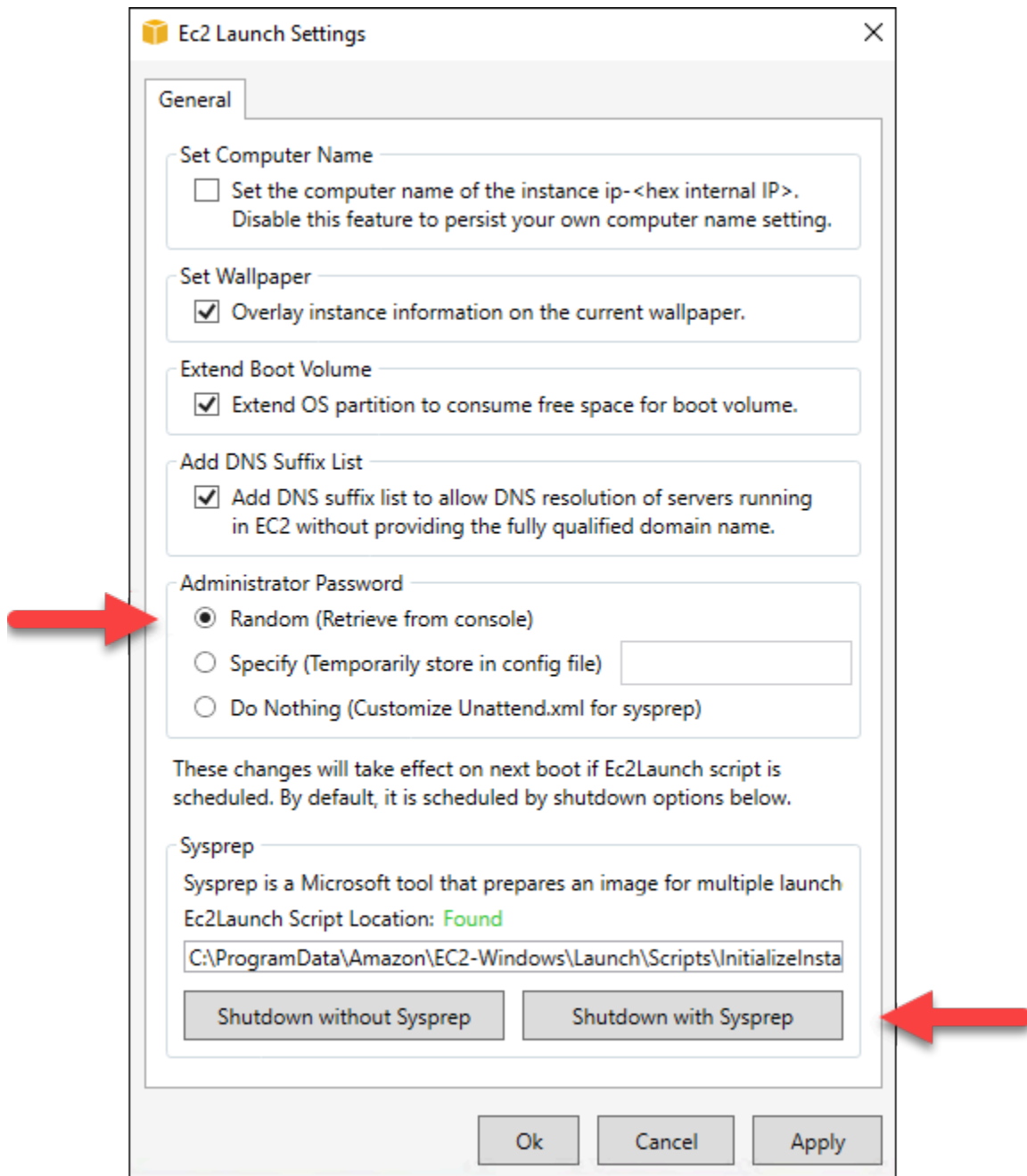
1. On the instance management page, choose the **Connect** tab, then choose **Connect using RDP**.

The browser-based RDP window opens, as shown in the following example:



2. On the taskbar, choose the Windows icon, or choose **Win** to open the Start menu.
3. Choose one of these options:
 - On Windows Server 2019 and Windows Server 2016 instances, choose **Start**, then choose **Ec2LaunchSettings**.
 - On Windows Server 2012 instances, choose **Start**, then choose **Ec2ConfigService Settings**.
4. In the Administrator Password section, choose **Random (Retrieve from console)**, then choose **Shutdown with Sysprep**.

On the Ec2ConfigService Settings application found in Windows Server 2012 instances, the **Random (Retrieve from console)** and **Shutdown with Sysprep** options are listed under the **Launch** tab.



5. Choose **Yes** to confirm that you want to run Sysprep and shut down the instance.

Your instance begins running Sysprep, your RDP connection shuts down, and your Lightsail instance stops running after a few minutes.

Step 3: Create a snapshot after running Sysprep

After your instance is in a stopped state, create a snapshot in the Lightsail console. When you create a snapshot of your Windows Server instance after running Sysprep, all instances that you

create based on the snapshot have a unique administrator password. You can connect to those instances by using the browser-based RDP client in the Lightsail console.

To create a snapshot in the Lightsail console

1. Toggle back to the Lightsail console.
2. On the instance management page for your Windows Server instance, choose the **Snapshots** tab
3. Under the **Manual snapshots** section of the page, choose **Create snapshot**, then enter a name for your snapshot.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
4. Choose **Create**.
 5. At the prompt, choose **Create snapshot** to confirm that you prepared the instance for the snapshot.

The snapshot process takes a few minutes to complete.

6. After the snapshot is created, choose **Start** at the top of the instance management page to start your instance again.

At this point, you should have two snapshots of your Windows Server instance as shown in the following example:



Use the Sysprep snapshot to create new instances. Use the backup snapshot only if the original instance doesn't function as expected after running Sysprep.

Next steps

Now that you have the Sysprep and backup snapshots, here are some next steps you should complete:

- Connect to your original instance, and confirm that your applications on it function as expected after running Sysprep. For more information, see [Connect to your Windows Server instance using Amazon Lightsail](#).
- Create a new instance using the Sysprep snapshot, connect to it, and confirm that your applications on the new instance function as expected. For more information, see [Create an instance from a snapshot](#).
- Delete your backup snapshot after you confirm that the original instance functions as expected after running Sysprep. For more information, see [Delete snapshots](#).
- If your instance doesn't function as expected after running Sysprep, then follow the steps in [Create an instance from a snapshot](#) to create a new instance from the backup snapshot.

Secure a Windows Server instance in Amazon EC2 created from a Lightsail snapshot

To improve the security of a Windows Server instance in Amazon Elastic Compute Cloud (Amazon EC2) created from an Amazon Lightsail snapshot, we recommend that you change the default administrator password. This removes the association between your Lightsail key pairs and your new Windows Server instance in Amazon EC2.

Note

If you created Linux or Unix instances in Amazon EC2 from a Lightsail snapshot, then you should perform a few steps to secure those instances. For more information, see [Secure an Amazon EC2 Linux or Unix instance that was created from a Lightsail snapshot](#).

Contents

- [Connect to your Windows Server instance in Amazon EC2](#)
- [Change the default administrator password of your Windows Server instance in Amazon EC2](#)

Connect to your Windows Server instance in Amazon EC2

To change your Windows Server administrator password, connect to your Windows Service instance in Amazon EC2 using Remote Desktop Protocol (RDP). To learn how to connect to your instance, see [Connect to a Windows Server instance in Amazon EC2 created from a Lightsail snapshot](#).

Continue to the [Change the default administrator password of your Windows Server instance in Amazon EC2](#) section of this guide after you're connected to your instance in Amazon EC2.

Change the default administrator password of your Windows Server instance in Amazon EC2

Change the default password on your Windows Server instance to remove the association between your Lightsail key pairs and your new Windows Server instance in Amazon EC2.

To change the default administrator password of your Windows Server instance in Amazon EC2

1. After you establish an RDP connection to your instance, open a Command Prompt and enter the following command.

```
net user Administrator "Password"
```

In the command, replace *Password* with your new password.

Example:

```
net user Administrator "%4=Bwk^GEAg8$u@5"
```

You should see a result similar to the following:

```
C:\Users\Administrator>net user Administrator "%4=Bwk^GEAg8$u@5"  
The command completed successfully.
```

```
C:\Users\Administrator>_
```

2. Store the new password in a safe place. You cannot retrieve the new password using the Amazon EC2 console. The console can retrieve only the default password. If you attempt to connect to the instance using the default password after changing it, an error message appears stating that your credentials did not work.

If you lose your password or it expires, you can generate a new password. For password reset procedures, see [Resetting a Lost or Expired Windows Administrator Password](#) in the Amazon EC2 documentation.

Secure a Linux or Unix instance in Amazon EC2 created from a Lightsail snapshot

Amazon Lightsail, and Amazon Elastic Compute Cloud (Amazon EC2), use public-key cryptography to encrypt and decrypt login information. Public-key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a key pair.

When you export a Linux or Unix Lightsail instance to EC2, the new EC2 instance will contain residual keys from the Lightsail service. As a security best practice, you should remove unused keys from your instance.

To improve the security of a Linux or Unix instance in EC2 that was created from a Lightsail snapshot, we recommend that you perform the following actions after creating the instance:

- Remove and replace the Lightsail default key if you used it to connect to the source instance in Lightsail. The Lightsail default key is not present in your Amazon EC2 instance if you used your own key to connect to your instance, or you created a key for your instance in the Lightsail console.
- Remove the Lightsail system key, also known as the `lightsail_instance_ca.pub` key. This key on Linux and Unix instances enables the Lightsail browser-based SSH client to connect. The `lightsail_instance_ca.pub` key is automatically removed when an EC2 instance is created using the **Create an Amazon EC2 instance** page in the Lightsail console or the Lightsail API.

Contents

- [Create a private key using Amazon EC2](#)
- [Create the public key using PuTTYgen](#)
- [Connect to your Linux or Unix instance in Amazon EC2](#)
- [Add the public key to your instance and test the connection](#)
- [Remove the Lightsail default key](#)

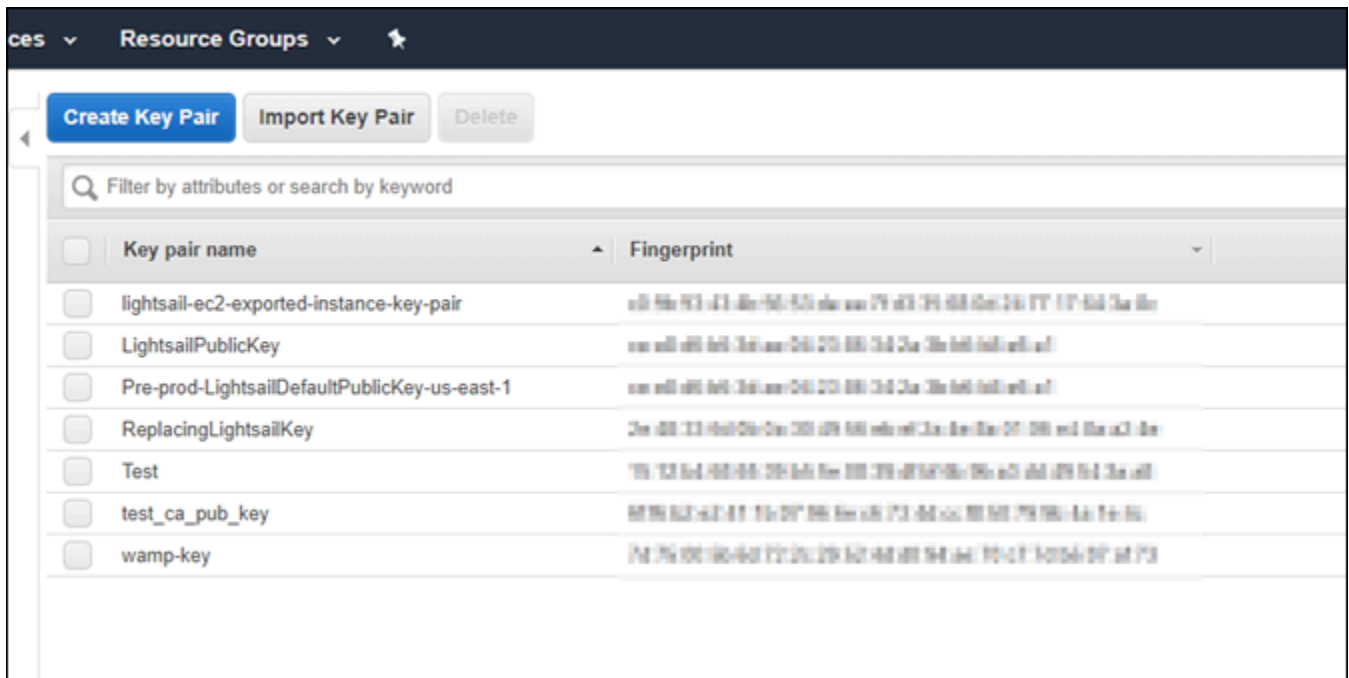
- [Remove the Lightsail system key](#)

Create a private key using Amazon EC2

Use the Amazon EC2 console to create a new key pair that you can use to replace the Lightsail default key pair.

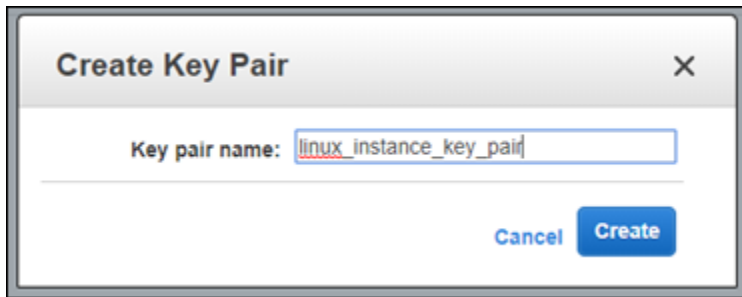
To create a private key using Amazon EC2

1. Sign in to the [Amazon EC2 console](#).
2. From the left navigation pane, choose **Key Pairs**.
3. Choose **Create key pair**.



4. Enter a name for the key into the **Key pair name** text box, then choose **Create**.

The new private key is automatically downloaded. Make note of where the private key is saved. You need it in the following *Create the public key using PuTTYgen* section of this guide to create a public key.



Create the public key using PuTTYgen

PuTTYgen is a tool that is included with PuTTY. Use PuTTYgen to generate the public key text that you add to your instance later in this guide.

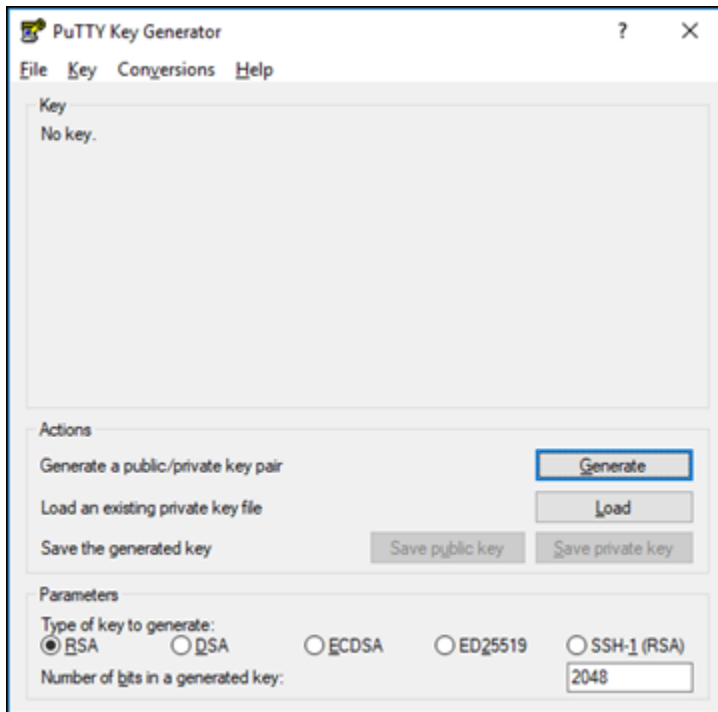
Note

For more information about how to configure PuTTY to connect to your Linux or Unix instance, see [Connect to an Amazon EC2 Linux or Unix instance that was created from a Lightsail snapshot](#).

To create the public key using PuTTYgen

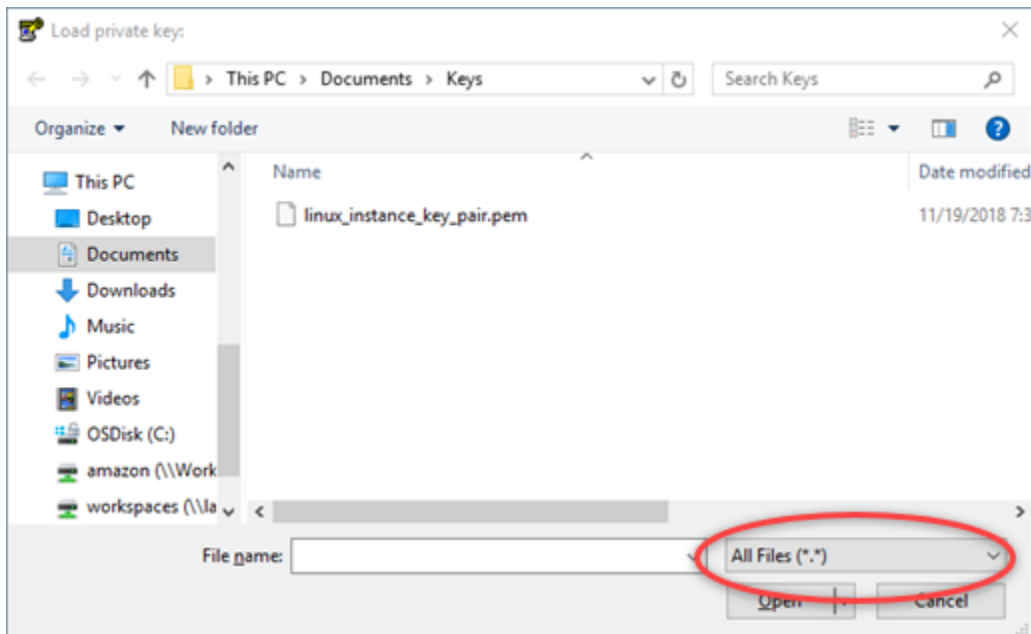
1. Start PuTTYgen.

For example, choose the **Windows Start** menu, choose **All Programs**, choose **PuTTY**, and choose **PuTTYgen**.



2. Choose **Load**.

By default, PuTTYgen displays only files with the .PPK extension. To locate your .PEM file, select the option to display files of all types.

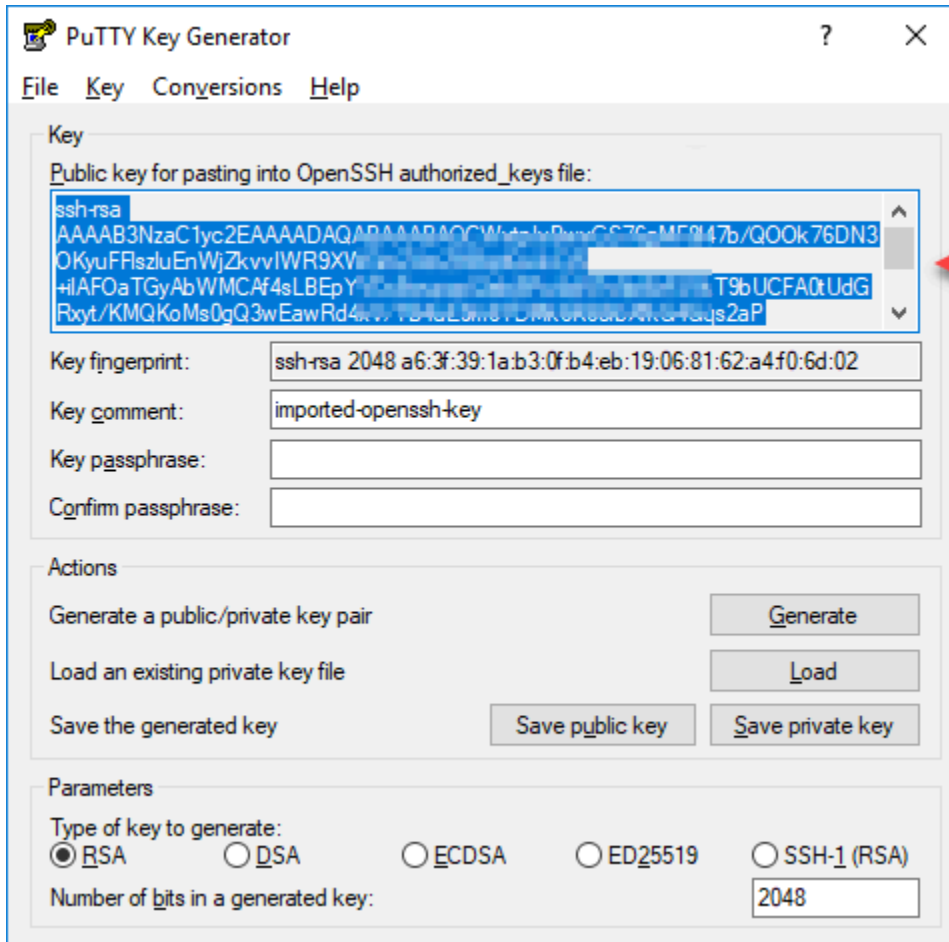


3. Navigate to the location of your private key that was created earlier in this guide. Choose the private key, and then choose **Open**.

4. After PuTTYgen confirms that you successfully imported the key, choose **OK**.

5. Highlight the contents of the **Public key** text box and copy it to your clipboard by pressing **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using macOS.

Open a text editor, such as Notepad or TextEdit, and paste the public key text into it by pressing **Ctrl+V** if you're using Windows, or **Cmd+V** if you're using macOS. Save the file with your public key text; you will need it later in this guide.



6. Continue to the [Connect to your Linux or Unix instance in Amazon EC2](#) section of this guide to connect to your EC2 instance and add the public key.

Connect to your Linux or Unix instance in Amazon EC2

Connect to your Linux or Unix instance in Amazon EC2 using SSH to remove the Lightsail default key and system key. For more information, see [Connect to a Linux or Unix instance in Amazon EC2 created from an Amazon Lightsail snapshot](#).

Continue to the [Add the public key to your instance and test the connection](#) section of this guide after you're connected to your instance in Amazon EC2.

Add the public key to your instance and test the connection

Public key content is saved in the `~/.ssh/authorized_keys` file on Linux and Unix instances. Edit the file to remove and replace the Lightsail default key from your Linux or Unix instance in Amazon EC2.

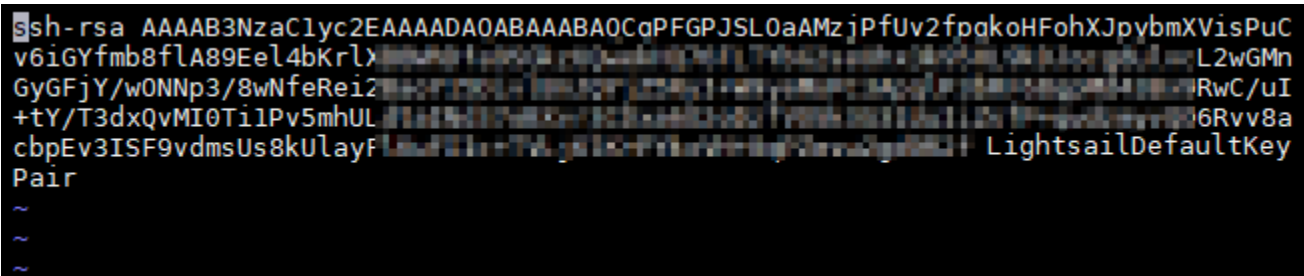
To add the public key to your instance and test the connection

1. After you establish an SSH connection to your instance, enter the following command to edit the `authorized_keys` file using the Vim text editor.

```
sudo vim ~/.ssh/authorized_keys
```

Note

These steps use Vim for demonstration purposes. However, you can use any text editor for these steps.



```
ssh-rsa AAAAB3NzaC1yc2EAAAADA0ABAAQAAOCqPFGPJSL0aAMzjPfUv2fpqkoHFohXJpybmXVisPuC  
v6iGYfmb8flA89Eel4bKrl> L2wGMn  
GyGFjY/wONnp3/8wNfeRei2 RWC/uI  
+tY/T3dxQvMI0Ti1Pv5mhUL 6Rvv8a  
cbpEv3ISF9vdmsUs8kUlayf LightsailDefaultKey  
Pair  
~  
~  
~
```

2. Press the `I` key to enter the insert mode in the Vim editor.
3. Enter an extra line after the Lightsail default key.
4. Copy and paste the public key text that you saved earlier in this guide.

The result should look like the following:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQcPFGPJSL0aAMzjPfuV2fpgkoHFohXJpybmXVisPuC
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQcWvtpIvBwvGS76gMF8l47b/Q00k76DN30KyUFFlszl
Pymgci5iWdhx1a8aDpgEvClwjsW+P9c7380Qny9PsUkiflymJE000Sb9czuR imported-openssh-key
```

Lightsail default key

New key

5. Press the ESC key, and then enter `:wq!` to save your edits, and quit Vim.
6. Enter the following command to restart the Open SSH server:

```
sudo /etc/init.d/sshd restart
```

You should see a result similar to the following:

```
[ec2-user@ip-172-26-11-173 ~]$ sudo /etc/init.d/sshd restart
Stopping sshd: [ OK ]
Starting sshd: [ OK ]
[ec2-user@ip-172-26-11-173 ~]$
```

Your new public key is now added to your instance. To test the new key pair, disconnect from your instance. Configure PuTTY to use your new private key instead of the Lightsail default key. If you're able to successfully connect to your instance using your new key pair, continue to the [Remove the Lightsail default key](#) section of this guide to remove the Lightsail default key.

Remove the Lightsail default key

Remove the Lightsail default key after you've added a new public key to your instance, and successfully connected to it using the new key pair.

To remove the Lightsail default key

1. After you establish an SSH connection to your instance, enter the following command to edit the `authorized_keys` file using the Vim text editor.

```
sudo vim ~/.ssh/authorized_keys
```

2. Press the I key to enter the insert mode in the Vim editor.
3. Delete the line that ends with `LightsailDefaultKeyPair`. This is the Lightsail default key.


```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQcPFGPJSL0aAMzjPfuV2fpgkoHFohXJpybmXVisPuC
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQcWvtpIvBwvGS76gMF8l47b/Q00k76DN30KyuFFlszl
Pymgci5iWdhx1a8aDpgEvClwjsW+P9c7380Qny9PsUkiflymJE000Sb9czuR imported-openssh-ke
y
~
~
```

Delete this line

Don't delete this line.
This is the new key.

4. Press the ESC key, and then enter `:wq!` to save your edits, and quit Vim.
5. Enter the following command to restart the Open SSH server:

```
sudo /etc/init.d/sshd restart
```

You should see a result similar to the following:

```
[ec2-user@ip-172-26-11-173 ~]$ sudo /etc/init.d/sshd restart
Stopping sshd: [ OK ]
Starting sshd: [ OK ]
[ec2-user@ip-172-26-11-173 ~]$
```

The Lightsail default key is now removed from your instance. Your instance will now refuse connections that use the Lightsail default key. Continue to the [Remove the Lightsail system key](#) section of this guide to remove the Lightsail system key.

Remove the Lightsail system key

The Lightsail system key, also known as the `lightsail_instance_ca.pub` key, on Linux and Unix instances enables the Lightsail browser-based SSH client to connect. Perform the following steps to remove the `lightsail_instance_ca.pub` key from your Linux or Unix instance in Amazon EC2, and edit the `/etc/ssh/sshd_config` file. The `/etc/ssh/sshd_config` file defines parameters for SSH connections to your instance.

To remove the Lightsail system key

1. In an SSH terminal window connected to your instance, enter the following command to remove the `lightsail_instance_ca.pub` key:

```
sudo rm -r /etc/ssh/lightsail_instance_ca.pub
```

2. Enter the following command to edit the `sshd_config` file using the Vim text editor.

```
sudo vim /etc/ssh/sshd_config
```

3. Press the I key to enter the insert mode in the Vim editor.
4. Delete the following text from the file, if it's present:

```
TrustedUserCAKeys /etc/ssh/lightsail_instance_ca.pub
```

5. Press the ESC key, and then enter `:wq!` to save your edits, and quit Vim.
6. Enter the following command to restart the Open SSH server:

```
sudo /etc/init.d/sshd restart
```

You should see a result similar to the following:

```
[ec2-user@ip-172-26-11-173 ~]$ sudo /etc/init.d/sshd restart
Stopping sshd: [ OK ]
Starting sshd: [ OK ]
[ec2-user@ip-172-26-11-173 ~]$
```

The `lightsail_instance_ca.pub` key is now removed from your instance. The associated `sshd_config` file is updated to exclude that key.

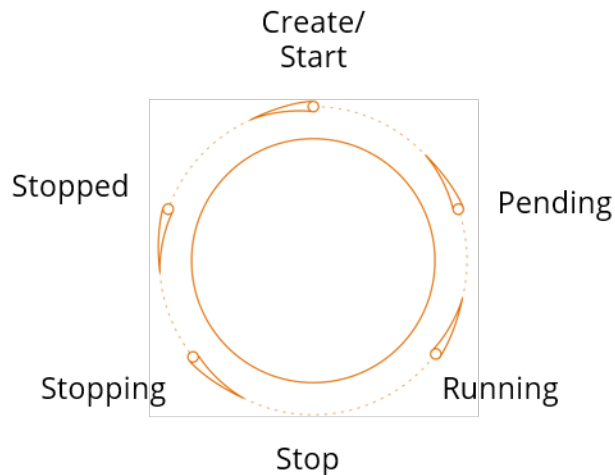
Manage your Lightsail instance

In Lightsail, your virtual private server is called an instance. You can connect to your instance, manage your ports and firewall settings, view metrics, associate a static IP with your instance, and more. Choose a task to learn how to get the most out of your instance:

- [Connect to your Linux or Unix instance](#)
- [View metrics](#)
- [Create a static IP address and attach it to an instance](#)
- [Firewall and ports](#)
- [Creating a snapshot of your Linux or Unix instance](#)
- [Start, stop, or restart your instance](#)
- [Force stop your instance](#)

Start, stop, or restart your Lightsail instance

When Lightsail creates your instance, your machine goes into a **Pending** state before it starts **Running**. After your instance is running, you can restart it or stop and then restart it. The cycle looks like this:



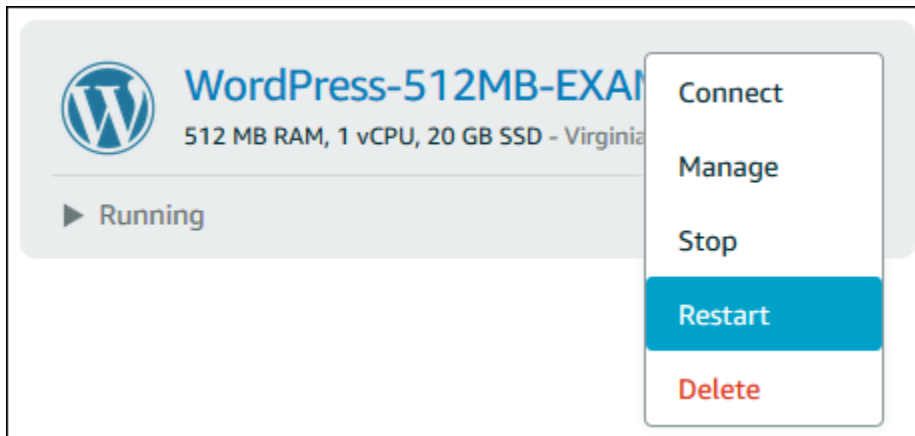
You can see the instance state when you manage your instance or view your instance on the home page.

⚠ Important

The default public IPv4 address that is assigned to your instance when you create it will change when you stop and start your instance. You can optionally create and attach a static IPv4 address to your instance. The static IPv4 address replaces the default public IPv4 address of your instance, and it stays the same when you stop and start your instance. For more information, see [Create a static IP and attach it to an instance](#).

Restart your instance while it's running

- On the home page, choose the instance you want to restart, or choose **Restart** from the manage instance menu.



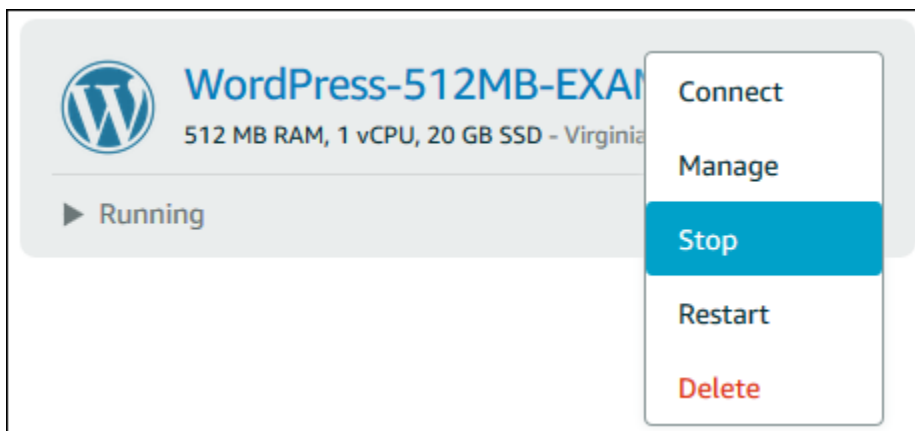
If you're viewing your instance from the instance management page, choose **Restart**, and then choose **Confirm** when prompted.

Note

To **Restart** your instance, it must be in a **Running** state.

Stop a running instance

- On the home page, choose the instance you want to stop, or choose **Stop** from the manage instance menu.



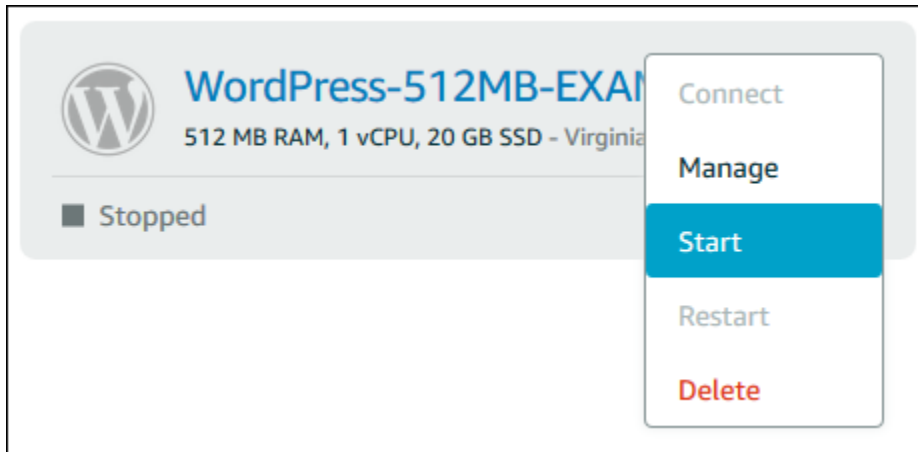
If you're viewing your instance from the instance management page, choose **Stop**, and then choose **Confirm** when prompted.

Note

To **Stop** your instance, it must be in a **Running** state.

Start your instance after it's stopped

- On the home page, choose the instance you want to start, or choose **Start** from the manage instance menu.



If you're viewing your instance from the instance management page, choose **Start**.

Note

To **Start** your instance, it must be in a **Stopped** state.

Update Amazon EC2 instances for enhanced networking

Some Lightsail instances are incompatible with the current generation EC2 instance types (T3, M5, C5, or R5) because they are not enabled for enhanced networking. If your source Lightsail instance is incompatible, you will need to choose a previous generation instance type (T2, M4, C4, or R4) when creating an EC2 instance from your exported snapshot. These instance type options are presented to you when creating an EC2 instance using the **Create an Amazon EC2 instance** page in the Lightsail console.

Note

For more information about enhanced networking, see [Enhanced Networking on Linux](#) or [Enhanced Networking on Windows](#) in the Amazon EC2 documentation.

To use the latest generation EC2 instance types when the source Lightsail instance is incompatible, you need to create the new EC2 instance using a previous generation instance type (T2, M4, C4, or R4), update the networking driver on your instance, and then upgrade the instance to the desired current generation instance type.

Prerequisites

You must create an Amazon EC2 instance from an exported Lightsail snapshot. If your Lightsail instance is incompatible, you'll choose a previous generation instance type (T2, M4, C4, or R4) when creating the Amazon EC2 instance. To learn more, see [Creating Amazon EC2 instances from exported snapshots in Lightsail](#).

After your new EC2 instance is up and running, continue to the [Enable Enhanced Networking with the Elastic Network Adapter](#) section of this guide to learn how to enable enhanced networking.

Enable Enhanced Networking with the Elastic Network Adapter

After your new instance is up and running, see one of the following guides in the Amazon EC2 documentation to enable enhanced networking with the Elastic Network Adapter (ENA):

- [Enabling Enhanced Networking with the ENA on Linux Instances](#)
- [Enabling Enhanced Networking with the ENA on Windows Instances](#)

Upgrade your instance type

After you have enabled enhanced networking, you can upgrade the instance type by following the instructions in one of the following guides:

- For Windows Server instances — [Migrating to Latest Generation Instance Types](#)
- For Linux or Unix instances — [Changing the Instance Type](#)

Extend the storage space of your Lightsail Windows Server instance

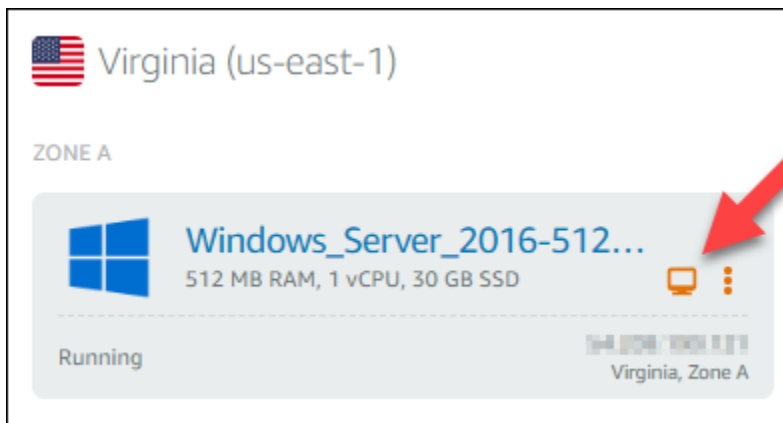
After you use a snapshot to create a new Windows Server instance with a larger plan, you may see that the available storage space is lower than that specified by the plan. This is typically because the additional storage space provided by the larger plan has not been allocated; therefore, it's not being used by the active volume. The steps in this topic show you how to extend the file system of your Windows Server instance to use the maximum storage space available.

Note

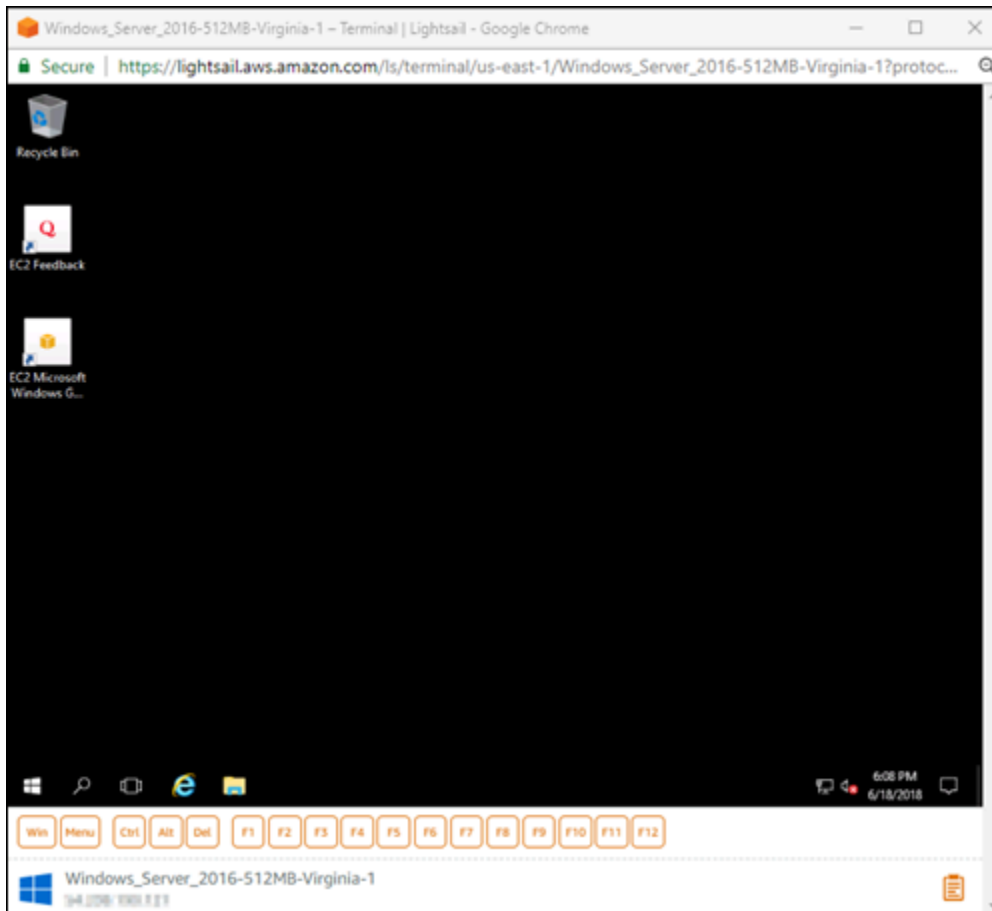
This scenario happens only when you create a Windows Server instance using a snapshot that was created before running the System Preparation (Sysprep) utility. For more information, see [Create a snapshot of your Windows Server instance](#).

To extend the file system for a Windows Server instance

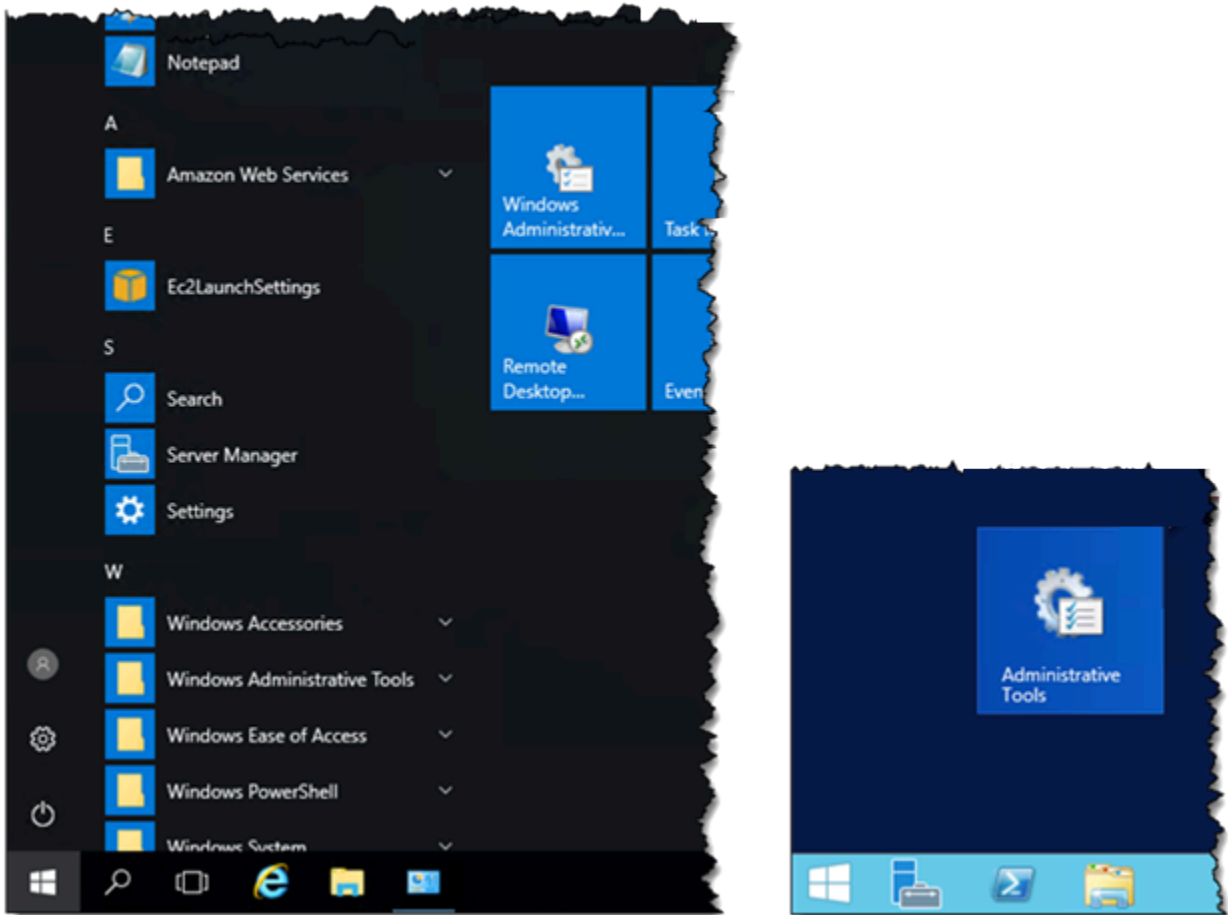
1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the RDP client icon for the instance you want to connect to.



The browser-based RDP client window opens, as shown in the following example:

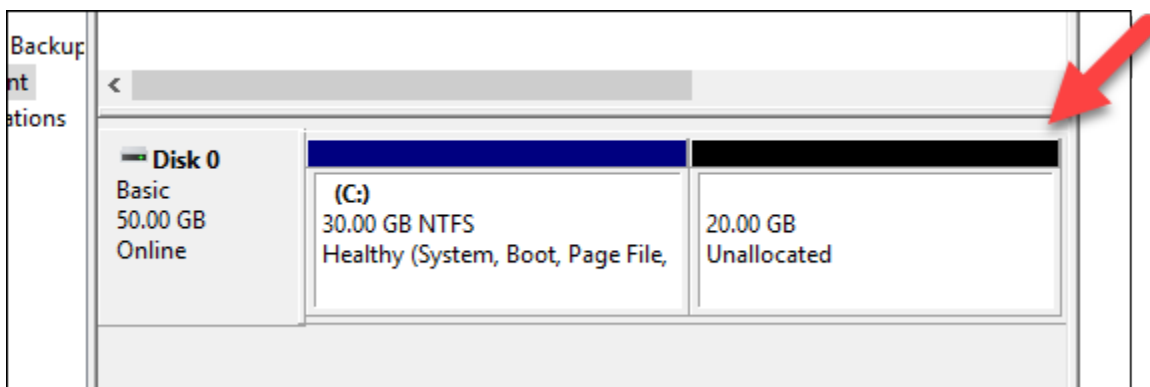


3. On the taskbar, choose the Windows icon, then choose one of the following options:
 - a. On Windows Server 2019 and Windows Server 2016 instances, choose **Start**, then choose **Windows Administrative Tools**.
 - b. On Windows Server 2012 instances, choose **Start**, then choose **Administrative Tools**.

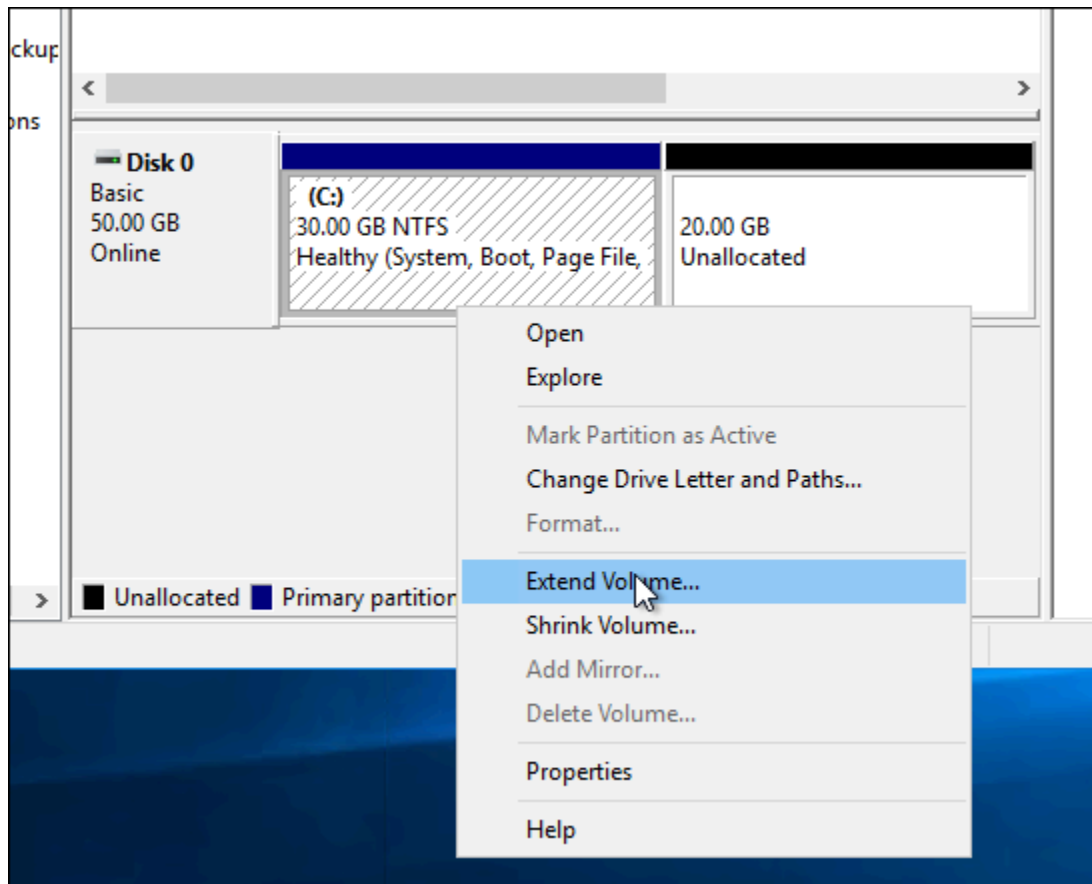


4. Choose **Computer Management**.
5. In the left pane of the Computer Management console, choose **Disk Management**.
6. On the **Actions** menu, choose **Rescan Disks**.

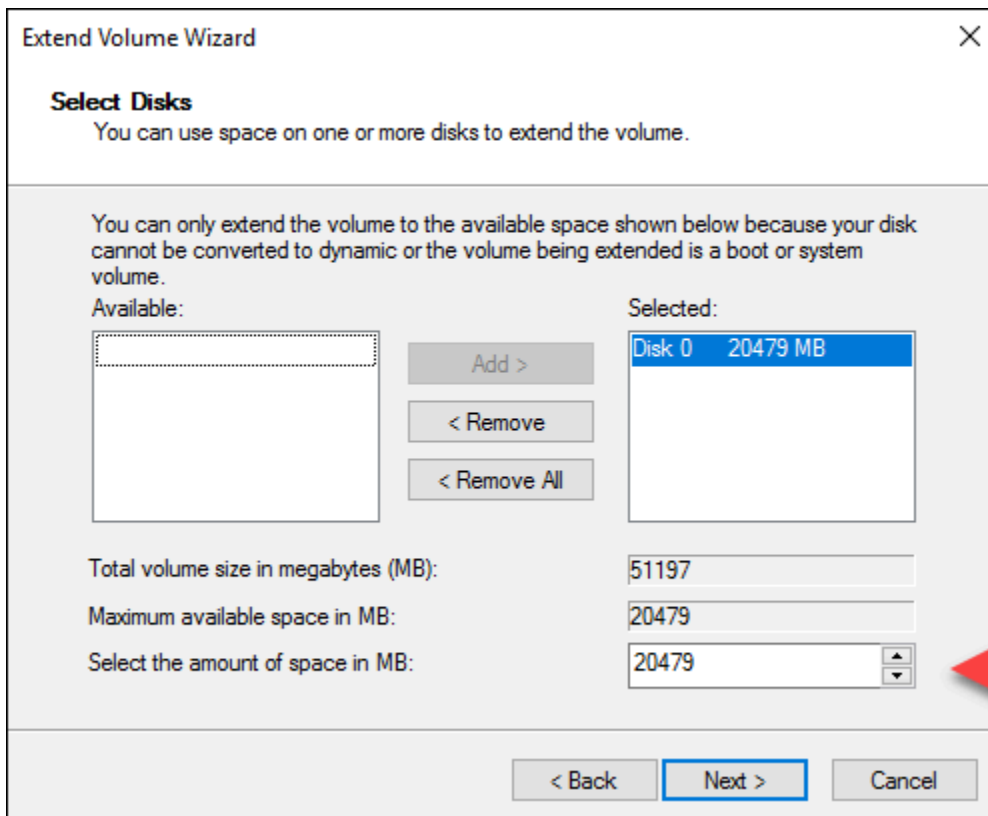
You may see unallocated space associated with a disk. Extend the active volume on the disk to use the unallocated space.



7. Right-click the active volume on the same disk as the unallocated space, then choose **Extend Volume**.

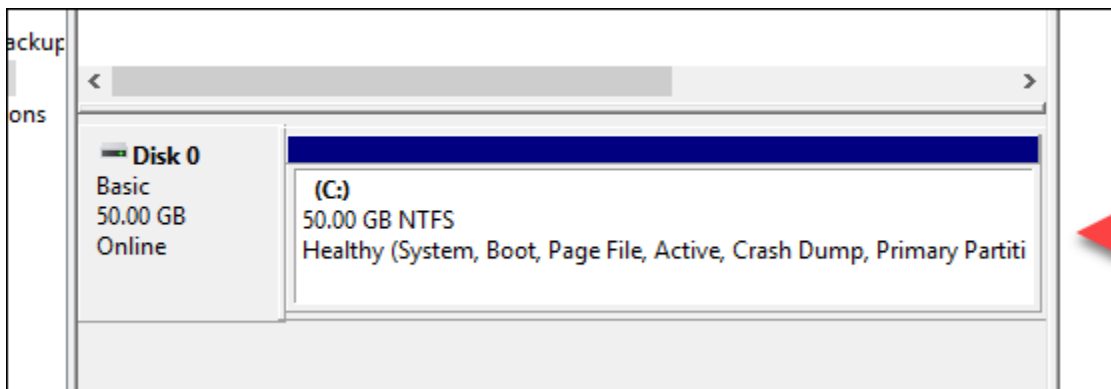


8. When the Extend Volume wizard opens, choose **Next**.
9. In the **Select the amount of space in MB** field, enter the number of megabytes by which to extend the volume. Normally, you set this to the maximum unallocated space. The value you enter is the amount of space that you are adding, not the final size of the volume.



10. Complete the Extend Volume wizard.

The active volume is extended to use the unallocated space that you specified. The following example shows all of the unallocated space chosen.



Use a launch script to configure your Lightsail instance when it starts up

When you create a Linux/Unix-based instance, you can add a launch script that does things like add software, update software, or configure your instance in some other way. To configure a Windows-

based instance with additional data, see [Configure your new Lightsail instance using Windows PowerShell](#).

Note

Depending on the machine image you choose, the command to get software on your instance varies. Amazon Linux uses `yum`, while Debian and Ubuntu both use `apt-get`. WordPress and other application images use `apt-get` because they run Ubuntu as their operating system. FreeBSD and openSUSE require additional user configuration to use custom tools such as `freebsd-update` or `zypper` (openSUSE).

Example: Configure an Ubuntu server to install Node.js

The following example updates the package list and then installs Node.js through the `apt-get` command.

1. On the **Create an instance** page, choose **Ubuntu** on the **OS Only** tab.
2. Scroll down and choose **Add launch script**.
3. Type the following:

```
# update package list
apt-get -y update
# install some of my favorite tools
apt-get install -y nodejs
```

Note

Commands you send to configure your server are run as root, so you don't need to include `sudo` before your commands.

4. Choose **Create instance**.

Example: Configure a WordPress server to download and install a plugin

The following example updates the package list, and then downloads and installs the [BuddyPress plugin](#) for WordPress.

1. On the **Create an instance** page, choose **WordPress**.
2. Choose **Add launch script**.
3. Type the following:

```
# update package list
apt-get -y update
# download wordpress plugin
wget "https://downloads.wordpress.org/plugin/buddypress.2.7.0.zip"
apt-get -y install unzip
# unzip into wordpress plugin directory
unzip buddypress.2.7.0.zip -d /var/wordpress/plugins
```

4. Choose **Create instance**.

Configure your Lightsail instance using Windows PowerShell or a batch script

When you create a Windows-based instance, you can configure it using a Windows PowerShell script or any other batch script. This is a one-time script that runs right after your instance launches. This topic shows the syntax of the scripts and provides an example to get you started. We also show you how to test your script to see if it ran successfully.

Create an instance that launches and runs a PowerShell script

The following procedure installs a tool called *chocolatey* on a new instance, right after the instance launches.

1. On the Lightsail home page, choose **Create instance**.
2. Choose the AWS Region and Availability Zone where you want to create your instance.
3. Under **Select a platform**, choose **Microsoft Windows**.
4. Choose **OS Only**, and then choose **Windows Server 2019**, **Windows Server 2016**, **Windows Server 2012 R2**.
5. Choose **Add launch script**.
6. Type the following:

```
<powershell>
```

```
iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))  
</powershell>
```

Note

You must always wrap your PowerShell scripts in `<powershell></powershell>` tags. You can enter non-PowerShell commands or batch scripts using `<script></script>` tags or without any tags at all.

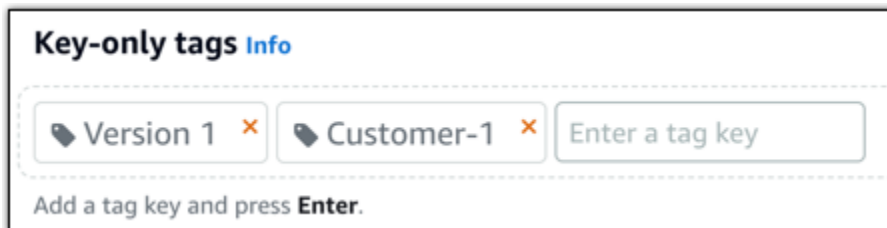
7. Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

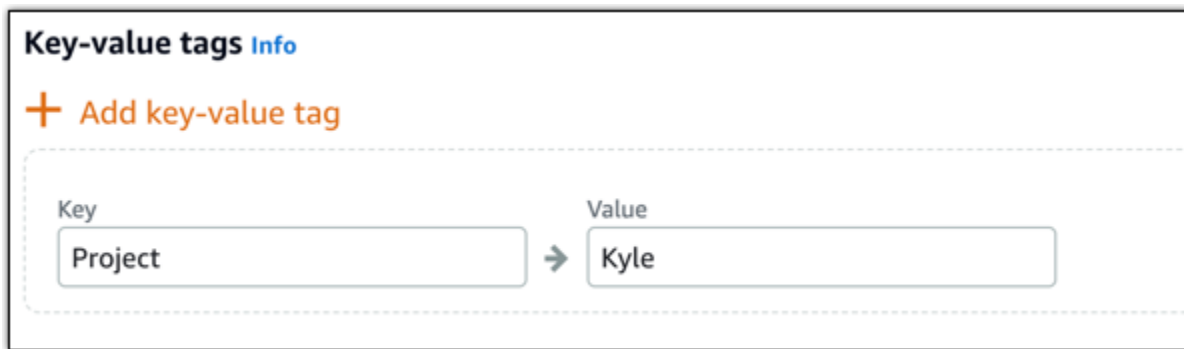
8. Choose one of the following options to add tags to your instance:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key: Project → Value: Kyle

Note

For more information about key-only and key-value tags, see [Tags](#).

9. Choose **Create instance**.

Verify that your script ran successfully

You can log in to your instance to verify that the script ran successfully. It can take up to 15 minutes for a Windows-based instance to be ready to accept RDP connections. Once it's ready, log in using the browser-based RDP client or configure your own RDP client. For more information, see [Connect to your Windows-based instance](#).

1. Once you can connect to your Lightsail instance, open a command prompt (or open Windows Explorer).
2. Change to the Log directory by typing the following:

```
cd C:\ProgramData\Amazon\EC2-Windows\Launch\Log
```

Note

In Windows Server 2012, the command is `cd C:\Program Files\Amazon\Ec2ConfigService\Logs`.

3. Open `UserdataExecution.log` in a text editor, or type the following: type `UserdataExecution.log`.

You should see the following in your log file.

```
2017/10/11 20:32:12Z: <powershell> tag was provided.. running powershell content
2017/10/11 20:32:13Z: Message: The output from user scripts: iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
2017/10/11 20:32:13Z: Userdata execution done
```

Best practices for securing Windows Server instances in Lightsail

In this article, we provide tips and tricks to help you avoid security risks when using your Lightsail instance running Windows Server.

About Lightsail passwords

When you create a Windows Server-based instance, Lightsail randomly generates a long password that is hard to guess. You use this password uniquely with your new instance. You can use the default password to connect quickly to your instance using remote desktop (RDP). You are always logged in as the **Administrator** on your Lightsail instance.

Manage your password

You can change the password on your Windows Server-based instance. This might be useful if you want to use a remote desktop client to access your Lightsail instance. Lightsail never stores a password you generate.

Note

You can use either the Lightsail-generated password or your own custom password with the browser-based RDP client in Lightsail. If you use a custom password, you will be prompted for your password every time you log in. It's easier to use the Lightsail-generated default password with the browser-based RDP client if you want quick access to your instance.

Use the Windows Server password manager to change your password securely. Press **Ctrl + Alt + Del**, and then choose **Change a password**. Be sure to keep a record of your password, because Lightsail doesn't store your password. If you need to retrieve your password, see the following: [Change the Administrator password for a Windows-based instance.](#)

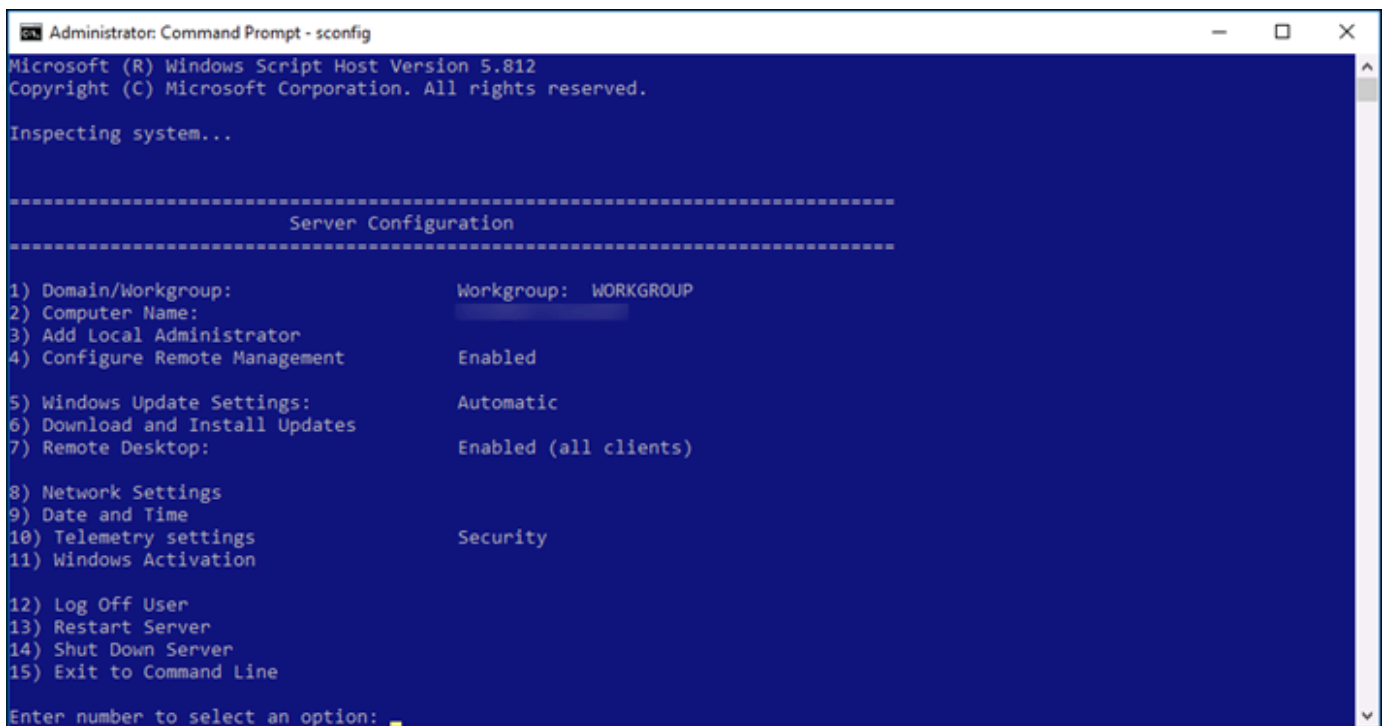
If you change your password from the unique, default password, be sure to use a strong password. You should avoid passwords that are based on names or dictionary words, or repeating sequences of characters.

Security patching

We recommend keeping your Windows Server-based Lightsail instances updated with the latest security patches. Be sure your server is configured to download and install updates. The following procedure tells you how to do this directly on your Lightsail instance running Windows Server.

1. On your Windows Server-based instance, open a command prompt.
2. Type `sconfig`, and then press Enter.

Windows Update Settings (number 5) are at Automatic by default.



```
Administrator: Command Prompt - sconfig
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

Inspecting system...

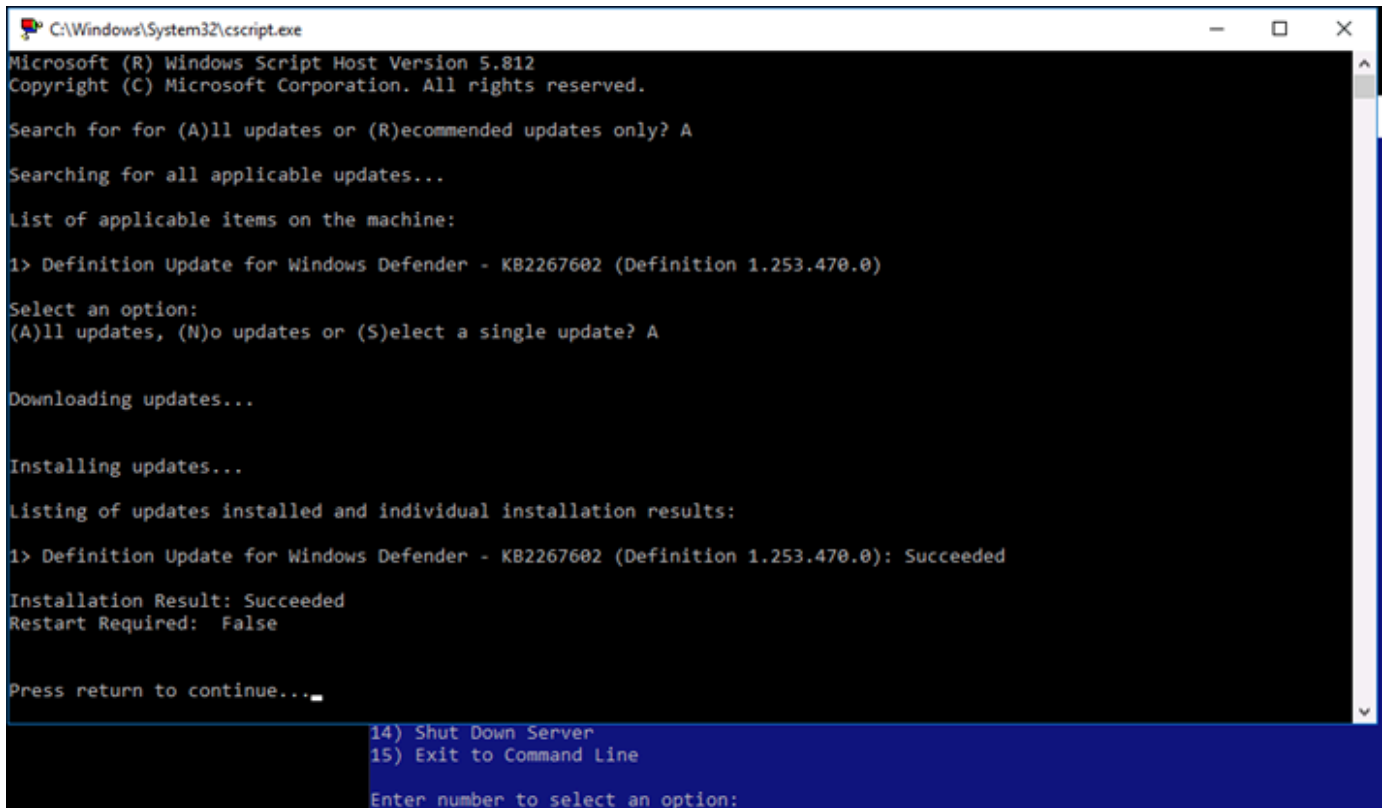
-----
Server Configuration
-----

1) Domain/Workgroup:           Workgroup:  WORKGROUP
2) Computer Name:
3) Add Local Administrator
4) Configure Remote Management  Enabled
5) Windows Update Settings:     Automatic
6) Download and Install Updates
7) Remote Desktop:             Enabled (all clients)
8) Network Settings
9) Date and Time
10) Telemetry settings         Security
11) Windows Activation
12) Log Off User
13) Restart Server
14) Shut Down Server
15) Exit to Command Line

Enter number to select an option:  
```

3. To download and install new updates, type 6, and then press Enter.
4. Type A to search for **(A)ll updates** in the new command window, and then press Enter.
5. Type A again to install **(A)ll updates**, and then press Enter.

When finished, you see a message with the installation results and more instructions (if those apply).



```
C:\Windows\System32\cmd.exe
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

Search for for (A)ll updates or (R)ecommended updates only? A
Searching for all applicable updates...

List of applicable items on the machine:
1> Definition Update for Windows Defender - KB2267602 (Definition 1.253.470.0)

Select an option:
(A)ll updates, (N)o updates or (S)elect a single update? A

Downloading updates...

Installing updates...

Listing of updates installed and individual installation results:
1> Definition Update for Windows Defender - KB2267602 (Definition 1.253.470.0): Succeeded

Installation Result: Succeeded
Restart Required: False

Press return to continue...

14) Shut Down Server
15) Exit to Command Line
Enter number to select an option:
```

Enable the Account Lockout Policy in Windows Server

You can configure Windows Server to temporarily or indefinitely disable accounts when a certain number of unsuccessful login attempts has been reached. For example, you can lock out someone who attempts to log in to your instance using three unsuccessful passwords.

For more information, see [Account Lockout Policy](#) in the *Windows Server documentation*.


Ports and firewall settings

By default, we open the following ports on your Windows Server-based instances.

Firewall ?

You can control which ports on this instance accept connections.

Application	Protocol	Port range
SSH	TCP	22
HTTP	TCP	80
RDP	TCP	3389



[+ Add another](#) [Edit rules](#) 



The ports you enable are exposed to the world and can't be restricted by source IP. To restrict access to your instance, you can turn off these ports and only enable them when you need to access your instance. Here's how:


1. Find the instance you want to manage in Lightsail, and then choose **Manage**.
2. Choose **Networking**.
3. On the **Networking** page for your instance, choose **Edit rules**.
4. Delete the RDP/TCP/3389 rule by choosing the orange "x" next to the rule.

Firewall ?

You can control which ports on this instance accept connections.

Application	Protocol	Port range	
HTTP	TCP	80	
RDP	TCP	3389	

[+ Add another](#) [Cancel](#)  [Save](#) 



5. Choose **Save**.

Lightsail firewall rules reference

You can add rules to an Amazon Lightsail instance's firewall that reflects the role of the instance. For example, an instance that's configured as a web server needs firewall rules that allow inbound HTTP and HTTPS access. A database instance needs rules that allow access for the type of

database, such as access over port 3306 for MySQL. For more information about firewalls, see [Instance firewalls in Lightsail](#).

This guide provides examples of the kinds of firewall rules that you can add to an instance firewall for specific kinds of access. The rules are listed as application, protocol, port, and source IP address (for example, application - protocol - port - source IP address), unless otherwise stated.

Contents

- [Web server rules](#)
- [Rules to connect to your instance from your computer](#)
- [Database server rules](#)
- [DNS server rules](#)
- [SMTP email](#)

Web server rules

The following inbound rules allow HTTP and HTTPS access.

Note

Some Lightsail instances have the following firewall rules configured by default. For more information, see [Firewalls and ports](#).

HTTP

HTTP - TCP - 80 - all IP addresses

HTTPS

HTTPS - TCP - 443 - all IP addresses

Rules to connect to your instance from your computer

To connect to your instance, you add a rule that allows SSH access (for Linux instances) or RDP access (for Windows instances).

Note

All Lightsail instances have either of the following firewall rules configured by default. For more information, see [Firewalls and ports](#).

SSH

SSH - TCP - 22 - The public IP address of your computer, or a range of IP addresses (in CIDR block notation) in your local network

RDP

RDP - TCP - 3389 - The public IP address of your computer, or a range of IP addresses (in CIDR block notation) in your local network

Database server rules

The following inbound rules are examples of rules that you might add for database access, depending on what type of database you're running on your instance.

SQL Server

Custom - TCP - 1433 - The public IP address of your computer, or a range of IP addresses (in CIDR block notation) in your local network

MySQL/Aurora

MySQL/Aurora - TCP - 3306 - The public IP address of your computer, or a range of IP addresses (in CIDR block notation) in your local network

PostgreSQL

PostgreSQL - TCP - 5432 - The public IP address of your computer, or a range of IP addresses (in CIDR block notation) in your local network

Oracle-RDS

Oracle-RDS - TCP - 1521 - The public IP address of your computer, or a range of IP addresses (in CIDR block notation) in your local network

Amazon Redshift

Custom - TCP - 5439 - The public IP address of your computer, or a range of IP addresses (in CIDR block notation) in your local network

DNS server rules

If you've set up your instance as a DNS server, you must ensure that TCP and UDP traffic can reach your DNS server over port 53.

DNS (TCP)

DNS (TCP) - TCP - 53 - The IP address of a computer, or a range of IP addresses (in CIDR block notation) in your local network

DNS (UDP)

DNS (UDP) - UDP - 53 - The IP address of a computer, or a range of IP addresses (in CIDR block notation) in your local network

SMTP email

To enable SMTP on your instance, you must configure the following firewall rule.

Important

After configuring the following rule, you must also configure reverse DNS for your instance. Otherwise, your email may be limited over TCP port 25. For more information, see [Configure reverse DNS for an email server](#).

SMTP

Custom - TCP - 25 - The IP addresses of the hosts that communicate with your instance

Instance firewalls in Amazon Lightsail

The firewall in the Amazon Lightsail console acts as a virtual firewall that controls the traffic allowed to connect to your instance through its public IP address. Each instance that you create

in Lightsail has two firewalls; one for IPv4 addresses and another for IPv6 addresses. Each firewall contains a set of rules that filter traffic coming into the instance. Both firewalls are independent of each other; you must configure firewall rules separately for IPv4 and IPv6. Edit your instance's firewall, at any time, by adding and deleting rules to allow or restrict traffic.

Contents

- [Lightsail firewalls](#)
- [Create firewall rules](#)
- [Specify protocols](#)
- [Specify ports](#)
- [Specify application layer protocol types](#)
- [Specify source IP addresses](#)
- [Default Lightsail firewall rules](#)
- [More information about firewalls](#)

Lightsail firewalls

Each Lightsail instance has two firewalls; one for IPv4 addresses and another for IPv6 addresses. All internet traffic into and out of your Lightsail instance passes through its firewalls. An instance's firewalls control the internet traffic that is allowed to flow into your instance. However, they don't control the traffic that flows out of it—the firewalls allow all outbound traffic. Edit your instance's firewalls, at any time, by adding and deleting rules to allow or restrict incoming traffic. Note that both firewalls are independent of each other; you must configure firewall rules separately for IPv4 and IPv6.

Firewall rules are always permissive; you can't create rules that deny access. You add rules to your instance's firewalls to allow traffic to reach your instance. When you add a rule to your instance's firewall, you specify the protocol to use, the port to open, and the IPv4 and IPv6 addresses that are allowed to connect to your instance, as shown in the following example (for IPv4). You can also specify an application layer protocol type, which is a preset that specifies the protocol and port range for you based on the service that you plan to use on your instance.

IPv4 Firewall [?](#)

Create rules to open ports to the internet, or to a specific IPv4 address or range.

[Learn more about firewall rules](#)

+ Add rule

Application	Protocol	Port or range / Code	Restricted to		
SSH	TCP	22	Any IPv4 address Lightsail browser SSH/RDP ?		
HTTP	TCP	80	Any IPv4 address		
HTTPS	TCP	443	Any IPv4 address		

Important

Firewall rules affect only traffic that flows in through the public IP address of an instance. It does not affect traffic that flows in through the private IP address of an instance, which can originate from Lightsail resources in your account, in the same AWS Region, or resources in a peered virtual private cloud (VPC), in the same AWS Region.

Firewall rules, and their configurable parameters are explained in the next few sections of this guide.

Create firewall rules

Create a firewall rule to enable a client to establish a connection with your instance, or with an application running on your instance. For example, to enable all web browsers to connect to the WordPress application on your instance, you configure a firewall rule that enables the Transmission Control Protocol (TCP) over port 80 from any IP address. If this rule is already configured on your instance's firewall, then you can delete it to block web browsers from being able to connect to the WordPress application on your instance.

Important

You can use the Lightsail console to add up to 30 source IP addresses at a time. To add up to 60 IP addresses at a time, use the Lightsail API, AWS Command Line Interface (AWS CLI), or an AWS SDK. This quota is enforced separately for IPv4 rules and IPv6 rules. For example, a firewall can have 60 inbound rules for IPv4 traffic and 60 inbound rules for IPv6

traffic. We recommend you consolidate individual IP addresses into CIDR ranges. For more information, see the [Specify source IP addresses](#) section of this guide.

You can also enable an SSH client to connect to your instance, to perform administrative tasks on the server, by configuring a firewall rule that enables TCP over port 22 only from the IP address of the computer that needs to establish a connection. In this case, you would not want to allow any IP address to establish an SSH connection to your instance; since doing so could lead to a security risk on your instance.

Note

The firewall rule examples described in this section may exist in your instance's firewall by default. For more information, see [Default firewall rules](#) later in this guide.

If there is more than one rule for a specific port, we apply the most permissive rule. For example, if you add a rule that allows access to TCP port 22 (SSH) from IP address 192.0.2.1. Then, you add another rule that allows access to TCP port 22 from everyone. As a result, everyone has access to TCP port 22.

Specify protocols

A protocol is the format in which data is transmitted between two computers. Lightsail allows you to specify the following protocols in a firewall rule:

- **Transmission Control Protocol (TCP)** is primarily used for establishing and maintaining a connection between clients and the application running on your instance, until the exchange of data is complete. It is a widely used protocol, and one which you might often specify in your firewall rules. TCP guarantees that no transmitted data is missing, and that all of the data that's sent makes it to the intended recipient. Its ideal use is for network applications that need high reliability, and for which transmission time is relatively less critical, such as web browsing, financial transactions, and text messaging. These use-cases will lose significant value if parts of the data is lost.
- **User Datagram Protocol (UDP)** is primarily used for establishing low-latency and loss-tolerating connections between clients and the application running on your instance. Its ideal use is for network applications in which perceived latency is critical, such as gaming, voice, and

video communications. These use-cases can suffer some data loss without adversely affecting perceived quality.

- **Internet Control Message Protocol (ICMP)** is primarily used to diagnose network communication issues, such as to determine if data is reaching its intended destination in a timely manner. Its ideal use is for the Ping utility, which you can use to test the speed of the connection between your local computer and your instance. It reports how long it takes data to reach your instance and come back to your local computer.

Note

When you add an ICMP rule to the IPv6 firewall of your instance using the Lightsail console, the rule is automatically configured to use ICMPv6. For more information, see [Internet Control Message Protocol for IPv6](#) on *Wikipedia*.

- **All** is used to allow all protocol traffic to flow into your instance. Specify this protocol when you're unsure which protocol to specify. This includes all internet protocols; not just the ones specified above. For more information, see [Protocol Numbers](#) on the *Internet Assigned Numbers Authority website*.

Specifying ports

Similar to physical ports on your computer, which allow your computer to communicate with peripherals like your keyboard and mouse, network ports serve as internet communications endpoints for your instance. When a computer seeks to connect with your instance, it will expose a port to establish the communication.

The ports that you can specify in a firewall rule can range from 0 to 65535. When you create a firewall rule to enable a client to establish a connection with your instance, you specify the protocol that will be used (covered earlier in this guide), and the port numbers through which the connection can be established. You can also specify the IP addresses that are allowed to establish a using the protocol and port; this is covered in the next section of this guide.

Here are some of the commonly used ports along with the services that use them:

- Data transfer over File Transfer Protocol (FTP) uses port 20.
- Command control over FTP uses port 21.
- Secure Shell (SSH) uses port 22.

- Telnet remote login service, and unencrypted text messages uses port 23.
- Simple Mail Transfer Protocol (SMTP) email routing uses port 25.

Important

To enable SMTP on your instance, you must also configure reverse DNS for your instance. Otherwise, your email might be limited over TCP port 25. For more information, see [Configuring reverse DNS for an email server on your Amazon Lightsail instance](#).

- Domain Name System (DNS) service uses port 53.
- Hypertext Transfer Protocol (HTTP) used by web browsers to connect to websites uses port 80.
- Post Office Protocol (POP3) used by email clients to retrieve email from a server uses port 110.
- Network News Transfer Protocol (NNTP) uses port 119.
- Network Time Protocol (NTP) uses port 123.
- Internet Message Access Protocol (IMAP) used to manage digital mail uses port 143.
- Simple Network Management Protocol (SNMP) uses port 161.
- HTTP Secure (HTTPS) HTTP over TLS/SSL used by web browsers to establish an encrypted connection to websites uses port 443.

For more information, see [Service Name and Transport Protocol Port Number Registry](#) on the *Internet Assigned Numbers Authority website*.

Specify application layer protocol types

You can specify an application layer protocol type when you create a firewall rule, which are presets that specify the rule's protocol and port range for you based on the service that you want to enable on your instance. This way, you don't have to search for the common protocol and ports to use for services like SSH, RDP, HTTP, and others. You can simply choose those application layer protocol types, and the protocol and port is specified for you. If you prefer to specify your own protocol and port, then you can choose the **Custom rule** application layer protocol type, which gives you control of those parameters.

Note

You can specify the application layer protocol type only by using the Lightsail console. You cannot specify the application layer protocol type using the Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs.

The following application layer protocol types are available in the Lightsail console:

- **Custom** – Choose this option to specify your own protocol and ports.
- **All protocols** – Choose this option to specify all protocols, and specify your own ports.
- **All TCP** – Choose this option to use the TCP protocol but you're unsure of which port to open. This enables the TCP over all ports (0-65535).
- **All UDP** – Choose this option to use the UDP protocol but you're unsure of which port to open. This enables the UDP over all ports (0-65535).
- **All ICMP** – Choose this option to specify all ICMP types and codes.
- **Custom ICMP** – Choose this option to use the ICMP protocol and define an ICMP type and code. For more information about ICMP types and codes, see [Control Messages](#) on *Wikipedia*.
- **DNS** – Choose this option when you want to enable DNS on your instance. This enables TCP and UDP over ports 53.
- **HTTP** – Choose this option when you want to enable web browsers to connect to a website that is hosted on your instance. This enables TCP over port 80.
- **HTTPS** – Choose this option when you want to enable web browsers to establish an encrypted connection to a website that is hosted on your instance. This enables TCP over port 443.
- **MySQL/Aurora** – Choose this option to enable a client to connect to a MySQL or Aurora database hosted on your instance. This enables TCP over port 3306.
- **Oracle-RDS** – Choose this option to enable a client to connect to an Oracle or RDS database hosted on your instance. This enables TCP over port 1521.
- **Ping (ICMP)** – Choose this option to enable your instance to respond to requests using the Ping utility. On the IPv4 firewall, this enables ICMP type 8 (echo) and code -1 (all codes). On the IPv6 firewall, this enables ICMP type 129 (echo reply) and code 0.
- **RDP** – Choose this option to enable an RDP client to connect to your instance. This enables TCP over port 3389.

- **SSH** – Choose this option to enable an SSH client to connect to your instance. This enables TCP over port 22.

Specify source IP addresses

By default, firewall rules allow all IP addresses to connect to your instance through the specified protocol and port. This is ideal for traffic such as web browsers over HTTP and HTTPS. However, this poses a security risk for traffic such as SSH and RDP, since you would not want to allow all IP addresses to be able to connect to your instance using those applications. For that reason, you can choose to restrict a firewall rule to an IPv4 or IPv6 address or range of IP addresses.

- **For the IPv4 firewall** - You can specify a single IPv4 address (for example, 203.0.113.1), or a range of IPv4 addresses. In the Lightsail console, the range can be specified using a dash (for example, 192.0.2.0-192.0.2.255) or in CIDR block notation (for example, 192.0.2.0/24). For more information about CIDR block notation, see [Classless Inter-Domain Routing](#) on *Wikipedia*.
- **For the IPv6 firewall** - You can specify a single IPv6 address (for example, 2001:0db8:85a3:0000:0000:8a2e:0370:7334), or a range of IPv6 addresses. In the Lightsail console, the IPv6 range can be specified using only CIDR block notation (for example, 2001:db8::/32). For more information about IPv6 CIDR block notation, see [IPv6 CIDR blocks](#) on *Wikipedia*.

Default Lightsail firewall rules

When you create a new instance, its IPv4 and IPv6 firewalls are preconfigured with the following set of default rules that allow basic access to your instance. The default rules are different depending on the type of instance that you create. These rules are listed as application, protocol, port, and source IP address (for example, application - protocol - port - source IP address).

AlmaLinux, Amazon Linux 2, Amazon Linux 2023, CentOS, Debian, FreeBSD, openSUSE, and Ubuntu (base operating systems)

SSH - TCP - 22 - all IP addresses

HTTP - TCP - 80 - all IP addresses

WordPress, Ghost, Joomla!, PrestaShop, and Drupal (CMS applications)

SSH - TCP - 22 - all IP addresses

HTTP - TCP - 80 - all IP addresses

HTTPS - TCP - 443 - all IP addresses

cPanel & WHM (CMS application)

SSH - TCP - 22 - all IP addresses

DNS (UDP) - UDP - 53 - all IP addresses

DNS (TCP) - TCP - 53 - all IP addresses

HTTP - TCP - 80 - all IP addresses

HTTPS - TCP - 443 - all IP addresses

Custom - TCP - 2078 - all IP addresses

Custom - TCP - 2083 - all IP addresses

Custom - TCP - 2087 - all IP addresses

Custom - TCP - 2089 - all IP addresses

LAMP, Django, Node.js, MEAN, GitLab, and Nginx (development stacks)

SSH - TCP - 22 - all IP addresses

HTTP - TCP - 80 - all IP addresses

HTTPS - TCP - 443 - all IP addresses

Magento (eCommerce application)

SSH - TCP - 22 - all IP addresses

HTTP - TCP - 80 - all IP addresses

HTTPS - TCP - 443 - all IP addresses

Redmine (project management application)

SSH - TCP - 22 - all IP addresses

HTTP - TCP - 80 - all IP addresses

HTTPS - TCP - 443 - all IP addresses

Plesk (hosting stack)

SSH - TCP - 22 - all IP addresses

HTTP - TCP - 80 - all IP addresses

HTTPS - TCP - 443 - all IP addresses

Custom - TCP - 53 - all IP addresses

Custom - UDP - 53 - all IP addresses

Custom - TCP - 8443 - all IP addresses

Custom - TCP - 8447 - all IP addresses

Windows Server 2022, Windows Server 2019, and Windows Server 2016

SSH - TCP - 22 - all IP addresses

HTTP - TCP - 80 - all IP addresses

RDP - TCP - 3389 - all IP addresses

SQL Server Express 2022, SQL Server Express 2019, and SQL Server Express 2016

SSH - TCP - 22 - all IP addresses

HTTP - TCP - 80 - all IP addresses

RDP - TCP - 3389 - all IP addresses

More information about firewalls

The following are some articles to help you manage firewalls in Lightsail.

- [Add and edit instance firewall rules](#)
- [Firewall rules reference](#)

Add and edit instance firewall rules in Amazon Lightsail

You can add rules to the IPv4 and IPv6 firewalls of your Amazon Lightsail instance to control the traffic that is allowed to connect to it. When you add a firewall rule, you can specify the application layer protocol type, protocol, ports, and the source IPv4 or IPv6 addresses that are allowed to connect to your instance. For more information about firewalls, see [Firewalls and ports](#).

Contents

- [Add and edit firewall rules](#)
- [Delete instance firewall rules](#)
- [More information about firewalls](#)

Add and edit instance firewall rules

Complete the following steps to add or edit firewall rules in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.
3. Choose the name of the instance for which you want to add or edit a firewall rule.
4. Choose the **Networking** tab on your instance's management page.

The **Networking** tab displays your instance's public and private IP addresses, and the configured IPv4 or IPv6 firewalls for your instance.

Note

The IPv6 firewall is displayed only if you have enabled IPv6 for the instance. For more information, see [Enable or disable IPv6](#).

5. Complete one of the following steps depending on whether the source IP for the rule is an IPv4 or IPv6 address:
 - To add an IPv4 firewall rule, scroll down to the **IPv4 Firewall** section of the page, and choose **Add rule**.
 - To add an IPv6 firewall rule, scroll down to the **IPv6 Firewall** section of the page and choose **Add rule**.

You can also choose **Edit** (pencil icon) next to an existing rule on either of the firewalls to edit it.

6. Choose an application layer protocol type in the **Application** drop-down menu.


When you choose an application layer protocol type, a set of protocol and port presets are specified for you. Example values are **Custom**, **All TCP**, **All UDP**, **Custom ICMP**, **SSH**, and **RDP**.

You can configure the following optional settings depending on the application layer protocol type you select:

- (Optional) If you choose the **Custom** option, then you can select a value in the **Protocol** drop-down menu. The available protocol values are **TCP** and **UDP**.

You can also enter a single port number or range of port numbers (for example, 7000-8000) in the **Port** field.

- (Optional) If you choose the **Custom ICMP** option, then you can specify an ICMP type in the **Type** field, and an ICMP code in the **Code** field. For more information about ICMP types and codes, see [Control Messages](#) on *Wikipedia*.

 **Note**

When you add an ICMP rule to the IPv6 firewall of your instance using the Lightsail console, the rule is automatically configured to use ICMPv6. For more information, see [Internet Control Message Protocol for IPv6](#) on *Wikipedia*.

- (Optional) Select **Restrict to IP address** to restrict access for the specified protocol and port to a specific IP address or range of IP addresses. Leave this option unselected to allow all IP addresses for the specified protocol and port.

You can enter a single IPv4 address (for example, 203.0.113.1), or a range of IPv4 addresses. The range can be specified using a dash (for example, 192.0.2.0-192.0.2.255) or in CIDR block notation (for example, 192.0.2.0/24). For more information about CIDR block notation, see [Classless Inter-Domain Routing](#) on *Wikipedia*.

- (Optional) If you choose the **SSH** or **RDP** application layer protocol type, and then choose **Restrict to IP address**, you can choose **Allow Lightsail browser SSH/RDP** to allow connection to your instance using the browser-based SSH and RDP clients available in the Lightsail console. Leave this option unselected to block access through those browser-based clients.

7. Choose **Create** to add the rule to the firewall.

The firewall rule is added after a few moments.

Delete instance firewall rules

Complete the following steps to delete instance firewall rule in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.
3. Choose the name of the instance for which you want to delete a firewall rule.
4. Choose the **Networking** tab on your instance's management page.
5. Complete one of the following steps depending on whether the source IP for the rule is an IPv4 or IPv6 address:
 - To delete an IPv4 firewall rule, scroll down to the **IPv4 Firewall** section of the page, and choose **Delete** (the trash icon) next to an existing rule to delete it.
 - To delete an IPv6 firewall rule, scroll down to the **IPv6 Firewall** section of the page, and choose **Delete** (the trash icon) next to an existing rule to delete it.

Important

Firewall rules affect only traffic that flows in through the public IP address of an instance. It does not affect traffic that flows in through the private IP address of an instance, which can originate from Lightsail resources in your account, in the same AWS Region, or resources in a peered virtual private cloud (VPC), in the same AWS Region. For example, if you delete the SSH rule (TCP port 22) from the instance firewall, other instances in the same Lightsail account, and in the same AWS Region, can continue to connect to it using SSH by specifying the private IP address of the instance.

The firewall rule is deleted after a few moments.

More information about firewalls

The following are some articles to help you manage firewalls in Lightsail.

- [Firewall and ports](#)
- [Firewall rules reference](#)

Instance Metadata Service (IMDS) and user data in Lightsail

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories, for example, hostname, events, and security groups. You can also use instance metadata to access user data that you specified when launching your instance. For example, you can specify parameters for configuring your instance, or include a simple script. Instances can also include dynamic data, such as an instance identity document that is generated when the instance is launched.

Important

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by authentication or cryptographic methods. Anyone who has direct access to the instance, and potentially any software running on the instance, can view its metadata. Therefore, you should not store sensitive data, such as passwords or long-lived encryption keys, as user data.

Use the Instance Metadata Service

You can access instance metadata from a running instance in Lightsail by using one of the following methods:

- Instance Metadata Service Version 1 (IMDSv1) – a request/response method
- Instance Metadata Service Version 2 (IMDSv2) – a session-oriented method

Important

Not all instance blueprints in Lightsail support IMDSv2. Use the `MetadataNoToken` instance metric to track the number of calls to the instance metadata service that are using IMDSv1. For more information, see [View instance metrics](#).

For more information about using IMDS, see [Configure the Instance Metadata Service \(IMDS\)](#).

Additional IMDS documentation

The following IMDS documentation is available in the *Amazon Elastic Compute Cloud User Guide for Linux Instances* and the *Amazon Elastic Compute Cloud User Guide for Windows Instances*:

Note

In Amazon EC2, instance blueprints are referred to as Amazon Machine Images (AMIs).

- For Linux instances:
 - [Configure the instance metadata options](#)
 - [Retrieve instance metadata](#)
 - [Work with instance user data](#)
 - [Retrieve dynamic data](#)
 - [Instance metadata categories](#)
 - [Example: AMI launch index value](#)
 - [Instance identity documents](#)
- For Windows instances:
 - [Configure the instance metadata options](#)
 - [Retrieve instance metadata](#)
 - [Work with instance user data](#)
 - [Retrieve dynamic data](#)
 - [Instance metadata categories](#)
 - [Example: AMI launch index value](#)
 - [Instance identity documents](#)

Configure the Instance Metadata Service (IMDS) in Lightsail

You can access instance metadata from a running instance by using one of the following methods:

- Instance Metadata Service Version 1 (IMDSv1) – a request/response method
- ~~Instance Metadata Service Version 2 (IMDSv2) – a session-oriented method~~

⚠ Important

Not all instance blueprints in Lightsail support IMDSv2. Use the `MetadataNoToken` instance metric to track the number of calls to the instance metadata service that are using IMDSv1. For more information, see [View instance metrics](#).

By default, you can use either IMDSv1 or IMDSv2, or both. The instance metadata service distinguishes between IMDSv1 and IMDSv2 requests based on whether a `PUT` or `GET` header, which is unique to IMDSv2, is present in any given request. For more information, see [Add defense in depth against open firewalls, reverse proxies, and SSRF vulnerabilities with enhancements to the EC2 Instance Metadata Service](#).

You can configure the instance metadata service on each instance so that local code or users must use IMDSv2. When you specify that IMDSv2 must be used, IMDSv1 no longer works. For more information, see [Configure the instance metadata options](#) in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.

To retrieve instance metadata, see [Retrieve instance metadata](#) in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.

ℹ Note

The examples in this section use the IPv4 address of the instance metadata service: `169.254.169.254`. If you are retrieving instance metadata for instances over the IPv6 address, make sure to enable and use the IPv6 address instead: `fd00:ec2::254`. The IPv6 address of the instance metadata service is compatible with IMDSv2 commands.

How Instance Metadata Service Version 2 works

IMDSv2 uses session-oriented requests. With session-oriented requests, you create a session token that defines the session duration, which can be a minimum of one second and a maximum of six hours. During the specified duration, you can use the same session token for subsequent requests. After the specified duration expires, you must create a new session token to use for future requests.

⚠ Important

Lightsail instances launched from Amazon Linux 2023 will have IMDSv2 configured by default.

The following examples use Linux and PowerShell shell script and IMDSv2 to retrieve the top-level instance metadata items. These examples do the following:

- Create a session token lasting six hours (21,600 seconds) by using the PUT request
- Store the session token header in a variable named TOKEN (on Linux) or token (on Windows)
- Request the top-level metadata items by using the token

Start by running the following commands:

• On Linux:

- First, generate a token with the following command.

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"``
```

- Then, use the token to generate top-level metadata items with the following command.

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
```

• On Windows:

- First, generate a token with the following command.

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

- Then, use the token to generate top-level metadata items with the following command.

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/meta-data/
```

After you create a token, you can reuse it until it expires. In the following examples, each command gets the ID of the blueprint (Amazon Machine Image (AMI)) that's used to launch the instance. The token from the previous example is reused. It is stored in `$TOKEN` (on Linux) or `$token` (on Windows).

- **On Linux:**

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/ami-id
```

- **On Windows:**

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -uri http://169.254.169.254/latest/meta-data/ami-id
```

When you use IMDSv2 to request instance metadata, the request must include the following:

- **A PUT request** – Use a PUT request to initiate a session to the instance metadata service. The PUT request returns a token that must be included in subsequent GET requests to the instance metadata service. The token is required to access metadata when using IMDSv2.
- **The token** – Include the token in all GET requests to the instance metadata service. When token usage is set to `required`, requests without a valid token or with an expired token receive a `401 - Unauthorized` HTTP error code. For information about changing the token usage requirement, see [update-instance-metadata-options](#) in the *AWS CLI Command Reference*.
- The token is an instance-specific key. The token is not valid on other instances and will be rejected if you attempt to use it outside of the instance on which it was generated.
- The PUT request must include a header that specifies the time to live (TTL) for the token, in seconds. The TTL can be specified to a maximum of six hours (21,600 seconds). The token represents a logical session. The TTL specifies the length of time that the token is valid and, therefore, the duration of the session.
- After a token expires, to continue accessing instance metadata, you must create a new session using another PUT request.
- You can choose to reuse a token or create a new token with every request. For a small number of requests, it might be easier to generate and immediately use a token each time you need to access the instance metadata service. But for efficiency, you can specify a longer duration for the token and reuse it instead of writing a PUT request every time you need to request

instance metadata. There is no practical limit on the number of concurrent tokens, with each representing its own session. IMDSv2 is, however, still constrained by normal instance metadata service connection and throttling limits. For more information, see [Query throttling](#) in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.

HTTP GET and HEAD methods are allowed in IMDSv2 instance metadata requests. PUT requests are rejected if they contain an X-Forwarded-For header.

By default, the response to PUT requests has a response hop limit (time to live) of 1 at the IP protocol level. If you need a larger hop limit, you can adjust it by using the `update-instance-metadata-options` command. For example, you might need a larger hop limit for backward compatibility with container services running on the instance. For more information, see [update-instance-metadata-options](#) in the *AWS CLI Command Reference*.

Transition to using Instance Metadata Service Version 2

Use of Instance Metadata Service Version 2 (IMDSv2) is optional. Instance Metadata Service Version 1 (IMDSv1) will continue to be supported indefinitely. If you choose to migrate to using IMDSv2, we recommend that you use the following tools and transition path.

Tools for helping with the transition to IMDSv2

If your software uses IMDSv1, use the following tools to help reconfigure your software to use IMDSv2.

- **AWS software:** The latest versions of the AWS SDKs and the AWS CLI support IMDSv2. To use IMDSv2, make sure that your instances have the latest versions of the AWS SDKs and the AWS CLI. For information about updating the AWS CLI, see [Installing, updating, and uninstalling the AWS CLI](#) in the *AWS Command Line Interface User Guide*. All Amazon Linux 2 software packages support IMDSv2.
- **Instance metric:** IMDSv2 uses token-backed sessions, while IMDSv1 does not. The `MetadataNoToken` instance metric tracks the number of calls to the instance metadata service that are using IMDSv1. By tracking this metric to zero, you can determine if and when all of your software has been upgraded to use IMDSv2. For more information, see [Viewing instance metrics in Amazon Lightsail](#).
- **Updates to Lightsail API operations and AWS CLI commands:** For existing instances, you can use the [update-instance-metadata-options](#) AWS CLI command (or the [UpdateInstanceMetadataOptions](#) API operation) to require the use of IMDSv2. The following

command is an example. Make sure you replace *InstanceName* with the name of your instance, and *RegionName* with the AWS Region your instance is in.

```
aws lightsail update-instance-metadata-options --region RegionName --instance-name InstanceName --http-tokens required
```

Recommended path to requiring IMDSv2 access

Using the preceding tools, we recommend that you follow this path for transitioning to IMDSv2:

Step 1: At the start

Update the AWS SDKs, the AWS CLI, and your software that uses role credentials on your instances to IMDSv2-compatible versions. For information about updating the AWS CLI, see [Upgrading to the latest version of the AWS CLI](#) in the *AWS Command Line Interface User Guide*.

Then, change your software that directly accesses instance metadata (in other words, that does not use an AWS SDK) by using the IMDSv2 requests.

Step 2: During the transition

Track your transition progress by using the instance metric `MetadataNoToken`. This metric shows the number of calls to the instance metadata service that are using IMDSv1 on your instances. For more information, see [View instance metrics](#).

Step 3: When everything is ready on all instances

Everything is ready on all instances when the instance metric `MetadataNoToken` records zero IMDSv1 usage. At this stage, you can require IMDSv2 use through the [update-instance-metadata-options](#) command. You can make these changes on running instances; you do not need to restart your instances.

Updating instance metadata options for existing instances is available only through the Lightsail API or the AWS CLI. It is currently not available in the Lightsail console. For more information, see [update-instance-metadata-options](#).

Additional IMDS documentation

The following IMDS documentation is available in the *Amazon Elastic Compute Cloud User Guide for Linux Instances* and the *Amazon Elastic Compute Cloud User Guide for Windows Instances*:

Note

In Amazon EC2, instance blueprints are referred to as Amazon Machine Images (AMIs).

- For Linux instances:
 - [Configure the instance metadata options](#)
 - [Retrieve instance metadata](#)
 - [Work with instance user data](#)
 - [Retrieve dynamic data](#)
 - [Instance metadata categories](#)
 - [Example: AMI launch index value](#)
 - [Instance identity documents](#)
- For Windows instances:
 - [Configure the instance metadata options](#)
 - [Retrieve instance metadata](#)
 - [Work with instance user data](#)
 - [Retrieve dynamic data](#)
 - [Instance metadata categories](#)
 - [Example: AMI launch index value](#)
 - [Instance identity documents](#)

Block storage disks in Amazon Lightsail

System disks offer the consistent and low-latency performance you need to run your workloads. With Lightsail disks, you can scale your usage up or down within minutes—and pay a low price for only what you provision.

You can select up to an 80 GB system disk on your Linux/Unix-based or Windows Server-based instance. See [Get started with Linux-based instances in Lightsail](#) or [Get started with Windows Server-based instances](#).

You can also add more storage to your virtual private server by creating additional block storage disks. See [Create and attach block storage disks to your Linux-based instance](#) or [Create and attach block storage disks to your Windows Server instance](#).

Block storage disks

Block storage is a storage architecture that manages data as "blocks". Each storage block (known as a "disk" in Lightsail) acts like an individual hard disk that you can attach to your server. In general, you can use additional block storage for applications or software that must separate out specific data from their core service, and to protect application data in case of a failure or other issue with your instance and boot storage disk.

Lightsail offers solid-state drives (SSD) for block storage. This type of block storage balances a reasonable price and good performance. It's intended to support the vast majority of workloads that run on Lightsail. Lightsail additional block storage disks offer consistent performance and the low latency needed for applications or software that frequently access stored data.

Note

For customers with applications that require sustained IOPS performance or high amounts of throughput per disk, or for customers running large databases like MongoDB, Cassandra, etc., we recommend using Amazon EC2 with GP2 or Provisioned IOPS SSD storage instead of Lightsail.

You can learn more about [Amazon EBS volumes](#) in the *Amazon EC2 User Guide*.

Disk Quotas

- 20,000 GB per Region.
- 16 TB per disk maximum, or 8 GB per disk minimum.
- Each instance can have up to 15 attached disks, and 1 boot volume disk.

Create and attach Lightsail block storage disks to your Linux-based instance

You can create and attach additional block storage disks for your Lightsail instances. After you create additional disks, you need to connect to your Linux/Unix-based Lightsail instance and format and mount the disk.

This topic shows you how to create a new disk and attach it using Lightsail. It also describes how to connect to your Linux/Unix-based instance using SSH, so that you can format and mount your attached disk.

If you have a Windows Server-based instance, see the following topic instead: [Create and attach block storage disks to your Windows Server instance](#).

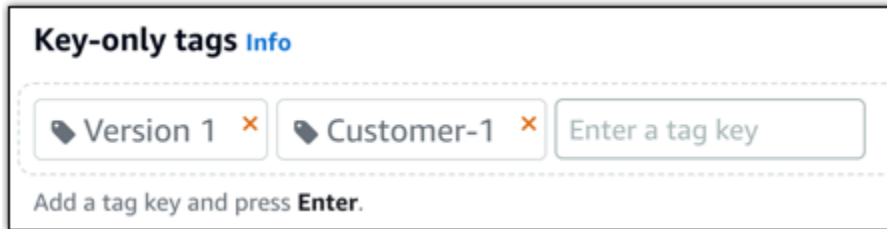
Step 1: Create a new disk and attach it to your instance

1. On the Lightsail home page, choose **Storage**.
2. Choose **Create disk**.
3. Choose the AWS Region and Availability Zone where your Lightsail instance is located.
4. Choose a size.
5. Enter a name for your disk.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
6. Choose one of the following options to add tags to your disk:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



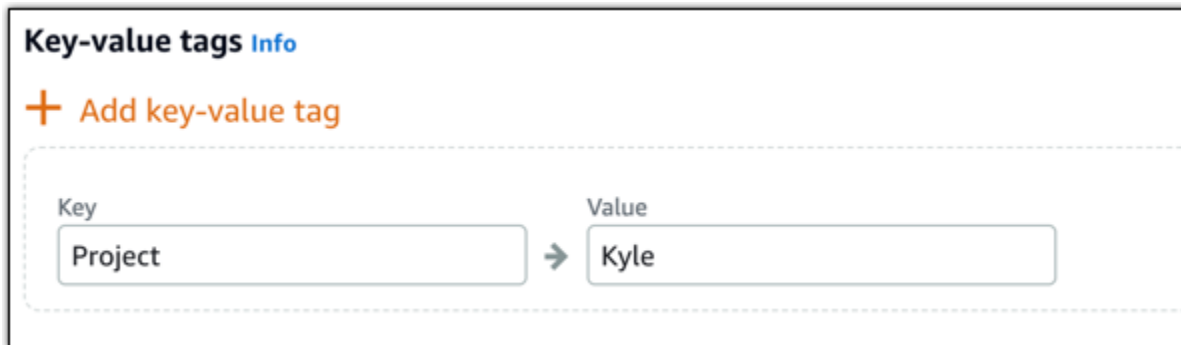
Key-only tags Info

Version 1 × Customer-1 × Enter a tag key

Add a tag key and press **Enter**.

- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key Value

Project → Kyle

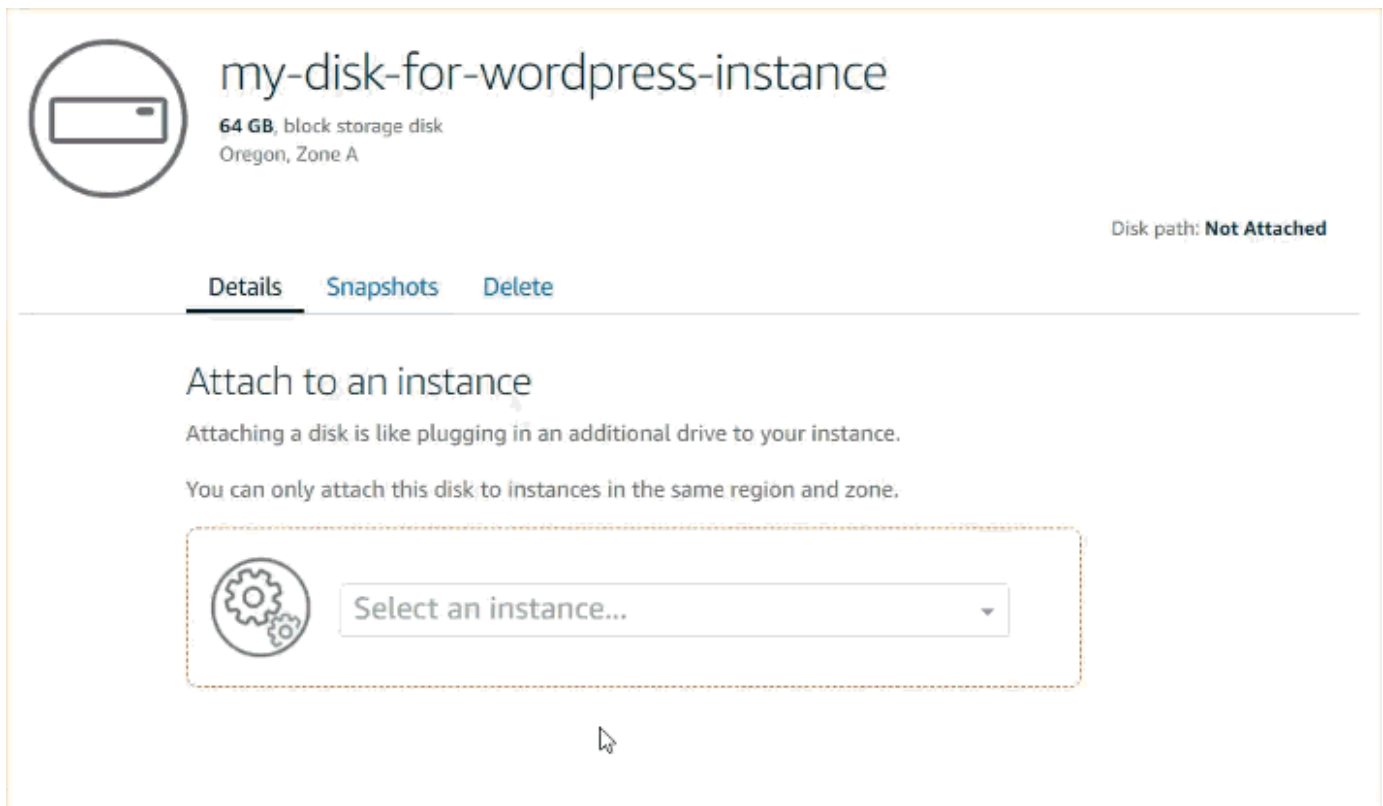
Note

For more information about key-only and key-value tags, see [Tags](#).

7. Choose **Create disk**.

After a few seconds, your disk is created and you're on the new disk management page.

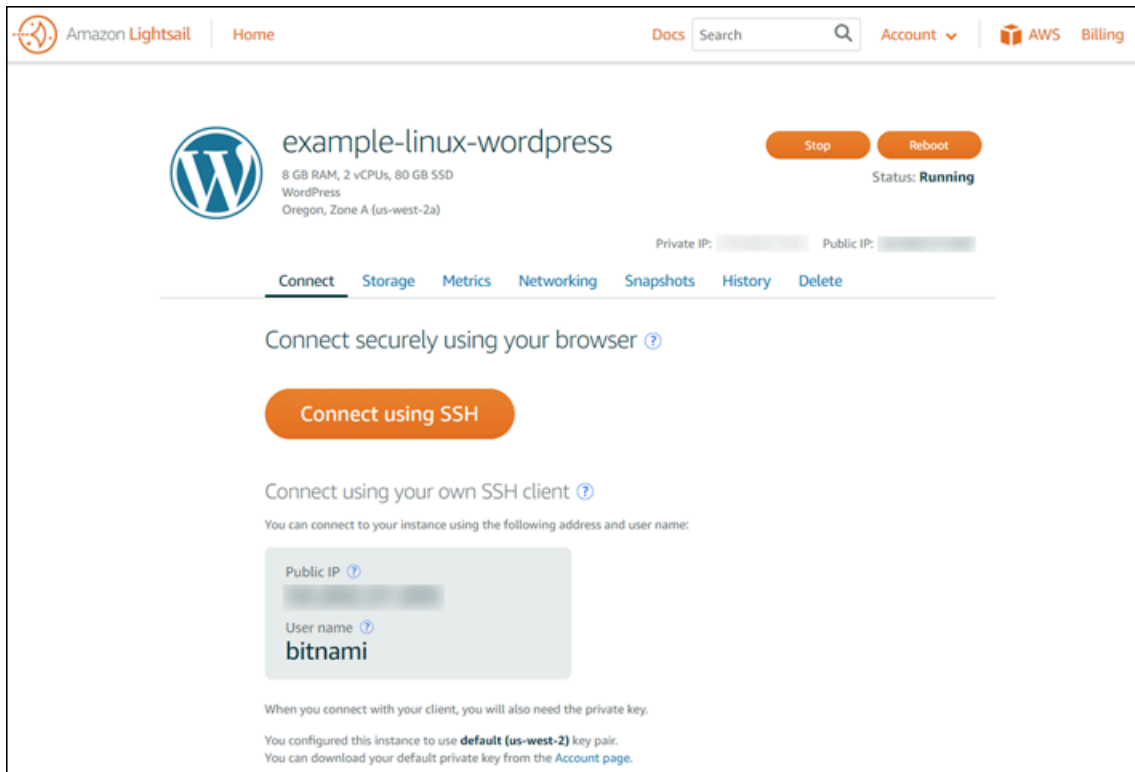
8. Choose your instance from the list, and then choose **Attach** to attach the new disk to your instance.



Step 2: Connect to your instance to format and mount the disk

1. After you create and attach your disk, go back to the instance management page in Lightsail.

The **Connect** tab is displayed by default.



2. Choose **Connect using SSH** to connect to your instance.
3. Type the following:

```
lsblk
```

You should see output like the following.

```
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda     202:0   0  80G  0 disk
##xvda1  202:1   0  80G  0 part /
xvdf     202:80   0  64G  0 disk
```

The output of `lsblk` removes the `/dev/` prefix from disk paths.

4. Determine whether to create a file system on the disk. New disks are raw block devices, and you must create a file system on them before you can mount and use them. Disks that have been restored from snapshots likely have a file system on them already. If you create a new file system on top of an existing file system, the operation overwrites your data. Use the following command to list special information, such as file system type.

```
sudo file -s /dev/xvdf
```

You should see the following output on a brand new disk.

```
/dev/xvdf: data
```

If you see output like the following, it means that your disk already has a file system.

```
/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=1701d228-e1bd-4094-a14c-12345EXAMPLE (needs journal recovery) (extents) (large files) (huge files)
```

5. Use the following command to create an ext4 file system on the disk. Substitute the device name (such as `/dev/xvdf`) for *device_name*. Depending on the requirements of your application or the limitations of your operating system, you can choose a different file system type, such as ext3 or XFS.

Important

This step assumes that you're mounting an empty disk. If you're mounting a disk that already has data on it (for example, a disk that was restored from a snapshot), don't use `mkfs` before mounting the disk. Instead, skip to step 6 of this procedure and create a mount point. Otherwise, you'll format the disk and delete the existing data.

```
sudo mkfs -t ext4 device_name
```

You should see output like the following.

```
mke2fs 1.42.9 (4-Feb-2014)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
4194304 inodes, 16777216 blocks
838860 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
512 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
```



```
Superblock backups stored on blocks:  
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,  
4096000, 7962624, 11239424
```

```
Allocating group tables: done  
Writing inode tables: done  
Creating journal (32768 blocks): done  
Writing superblocks and filesystem accounting information: done
```

6. Use the following command to create a mount point directory for the disk. The mount point is where the disk is located in the file system tree and where you read files from and write files to after you mount the disk. Substitute a location for *mount_point*, such as `/data`.

```
sudo mkdir mount_point
```

7. You can verify that the disk now has a file system on it by typing the following command.

```
sudo file -s /dev/xvdf
```

Instead of `/dev/xvdf`: `data`, you'll see output similar to the following.

```
/dev/xvdf: Linux rev 1.0 ext4 filesystem data, UUID=0ee83fdf-e370-442e-  
ae38-12345EXAMPLE (extents) (large files) (huge files)
```

8. Finally, mount the disk by typing the following command.

```
sudo mount device_name mount_point
```

Review the file permissions of your new disk mount to ensure that your users and applications can write to the disk. For more information about file permissions, see [Making an Amazon EBS Volume Available for Use](#) in the *Amazon EC2 User Guide*.

Step 3: Mount the disk every time you reboot your instance

You probably want to mount this disk every time you reboot your Lightsail instance. If you don't, this step is optional for you.

1. To mount this disk on every system reboot, add an entry for the device to the `/etc/fstab` file.

Create a backup of your `/etc/fstab` file that you can use if you accidentally destroy or delete this file while you're editing it.

```
sudo cp /etc/fstab /etc/fstab.orig
```

2. Open the `/etc/fstab` file using any text editor, such as vim.

You need to type `sudo` before opening the file so that you can save changes.

3. Add a new line to the end of the file for your disk using the following format.

```
device_name mount_point file_system_type fs_mntops fs_freq fs_passno
```

For example, your new line might look something like this.

```
/dev/xvdf /data ext4 defaults,nofail 0 2
```

4. Save the file and exit your text editor.

Create and attach Lightsail block storage disks to your Windows Server instance

If you need additional storage space, you can create and attach block storage disks to your Windows Server instance in Amazon Lightsail. For more information about block storage disks, see [Block storage disks](#).

This guide shows you how to create a new block storage disk and attach it to your Windows Server instance using the Lightsail console. It also describes how to connect to your Windows Server instance using RDP so that you can bring the disk online and initialize it.

This procedure is the same in Windows Server 2016 and Windows Server 2012 R2.

Note

If you have a Linux or Unix instance, see [Create and attach disks to your Linux or Unix instance](#).

Step 1: Create a new block storage disk and attach it to your instance

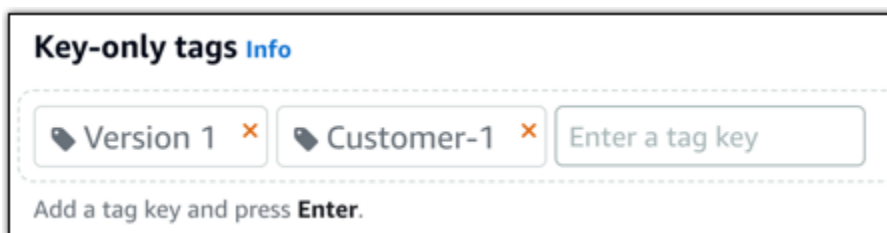
Create a new block storage disk and attach it to your instance using the Amazon Lightsail console.

To create a new block storage disk and attach it to your instance

1. Sign in to the [Lightsail console](#).
2. Choose the **Storage** tab, then choose **Create disk**.
3. Choose the AWS Region and Availability Zone where your Lightsail instance is located.
4. Choose a disk size.
5. Enter a name for your storage disk.

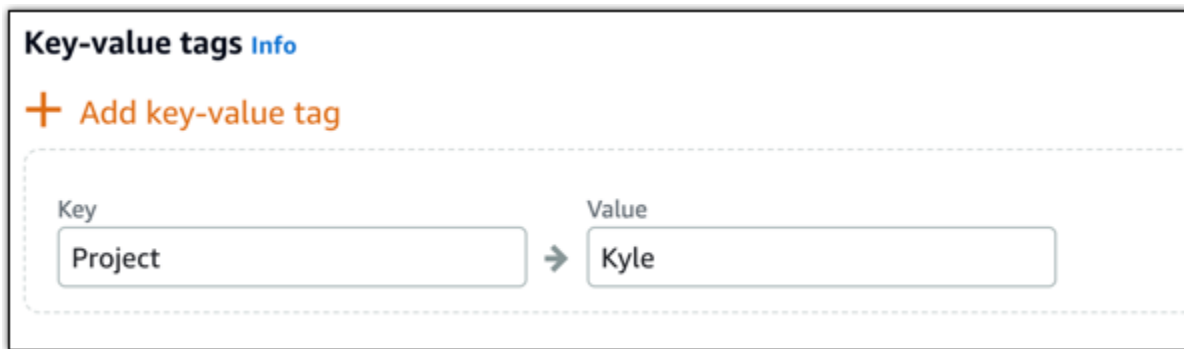
Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
6. Choose one of the following options to add tags to your disk:
 - **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.

**Note**

For more information about key-only and key-value tags, see [Tags](#).

7. Choose Create disk.

After a few seconds, the disk is created and you can view information about it on the disk management page.

8. Choose your instance from the list, and then choose Attach to attach the new disk to your instance.

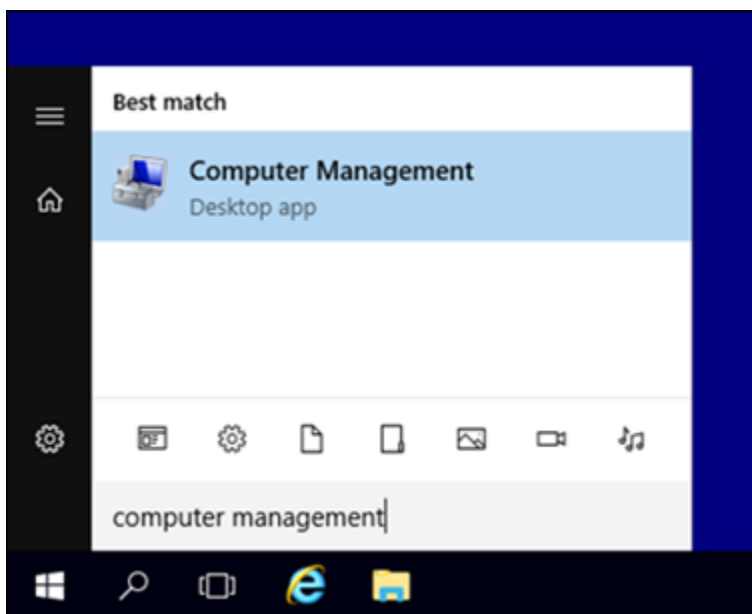
Continue to the [Step 2: Connect to your instance and bring the block storage disk online](#) section of this guide to bring the block storage disk online.

Step 2: Connect to your instance and bring the block storage disk online

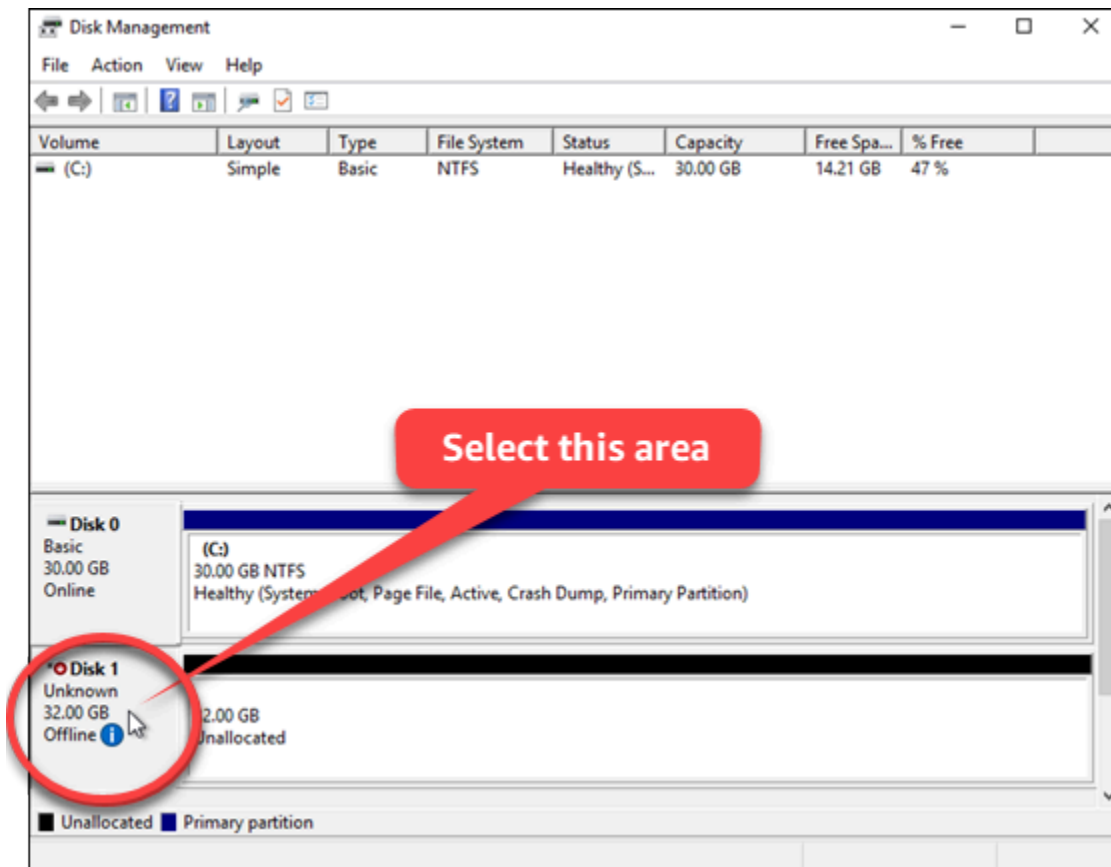
Connect to your Windows Server instance and use the Disk Management utility to bring the recently attached block storage disk online.

To connect to your instance and bring the block storage disk online

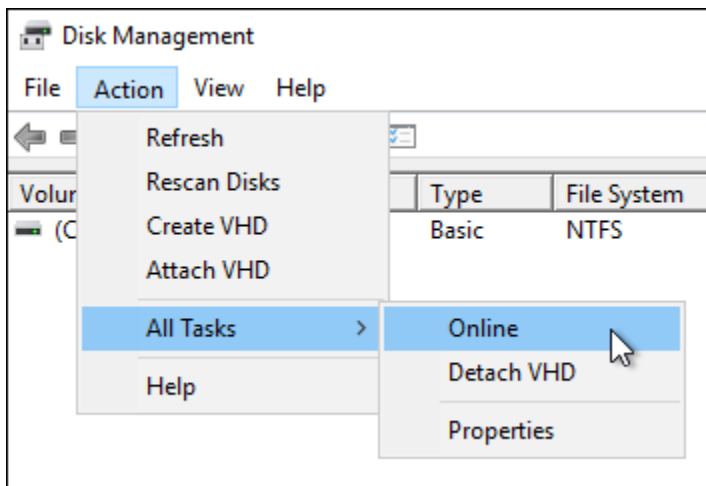
1. Navigate to the [Lightsail console home page](#).
2. Choose the name of the instance to which you attached the additional storage disk earlier in this guide.
3. Under the **Connect** tab, choose **Connect using RDP**.
4. On the Windows Start menu, search for **Computer Management**, and in the search results, choose **Computer Management**.



5. In Computer Management, in the left pane, choose **Disk Management**.
6. In the bottom pane of the Disk Management utility, select the disk labeled as **Unknown / Offline**. This is the block storage disk that you attached to your instance earlier in this guide.



7. With the disk selected, on the **Action** menu, choose **All Tasks**, and then choose **Online**.



You should see the status of the block storage disk update to **Not Initialized**. The block storage disk is not yet online. Continue to the [Step 3: Initialize the block storage disk](#) section of this guide to initialize the block storage disk.

Step 3: Initialize the block storage disk

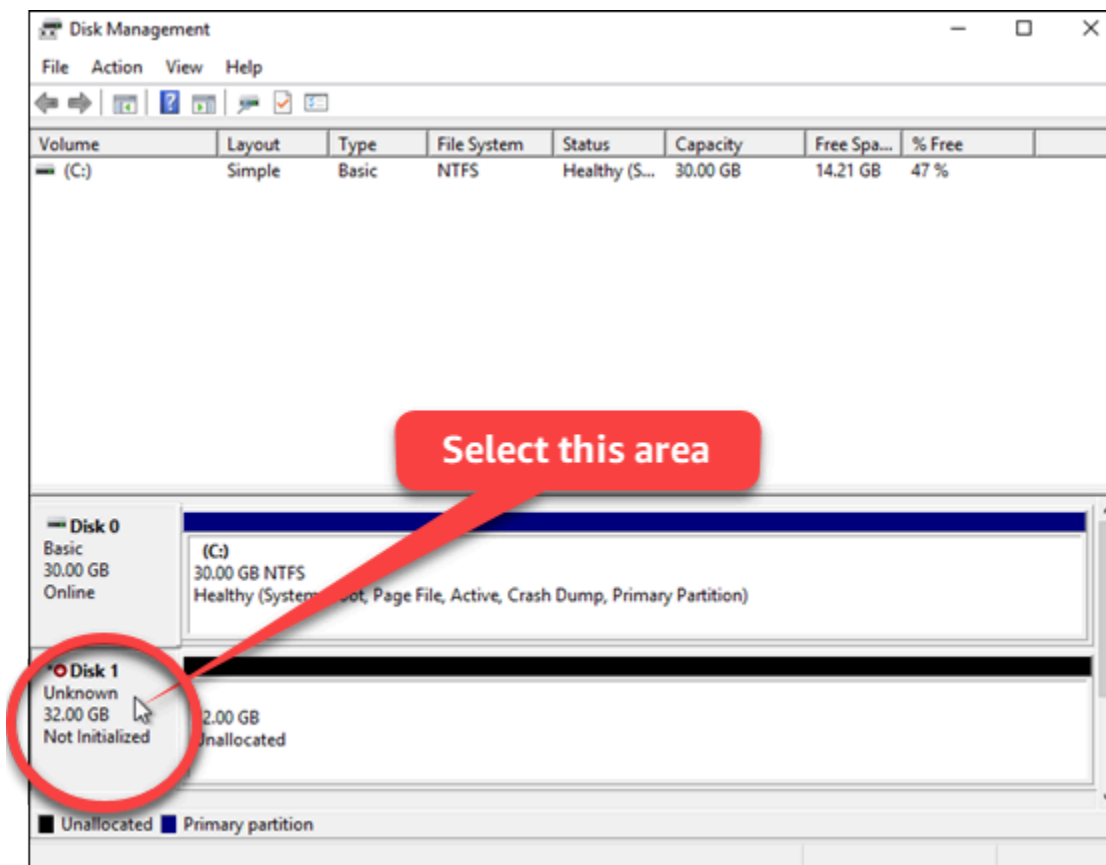
Initialize the block storage disk so that you can format it.

⚠ Important

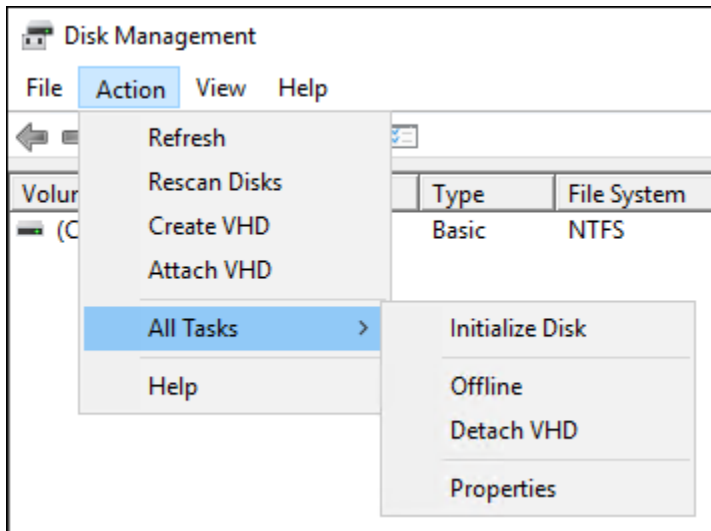
If you're mounting a disk that already has data on it, such as a disk that you created from a snapshot, be sure that you don't reformat the disk and delete the existing data.

To initialize the block storage disk

1. In the bottom pane of the Disk Management utility, select the disk labeled as **Unknown / Not initialized**.



2. With the disk selected, on the **Action** menu, choose **All Tasks**, and then choose **Initialize Disk**.



3. Choose the partition style for your new disk, and then choose **OK**.

Note

For more information about partition styles, see the [About partition styles - GPT and MBR](#) article from Microsoft.

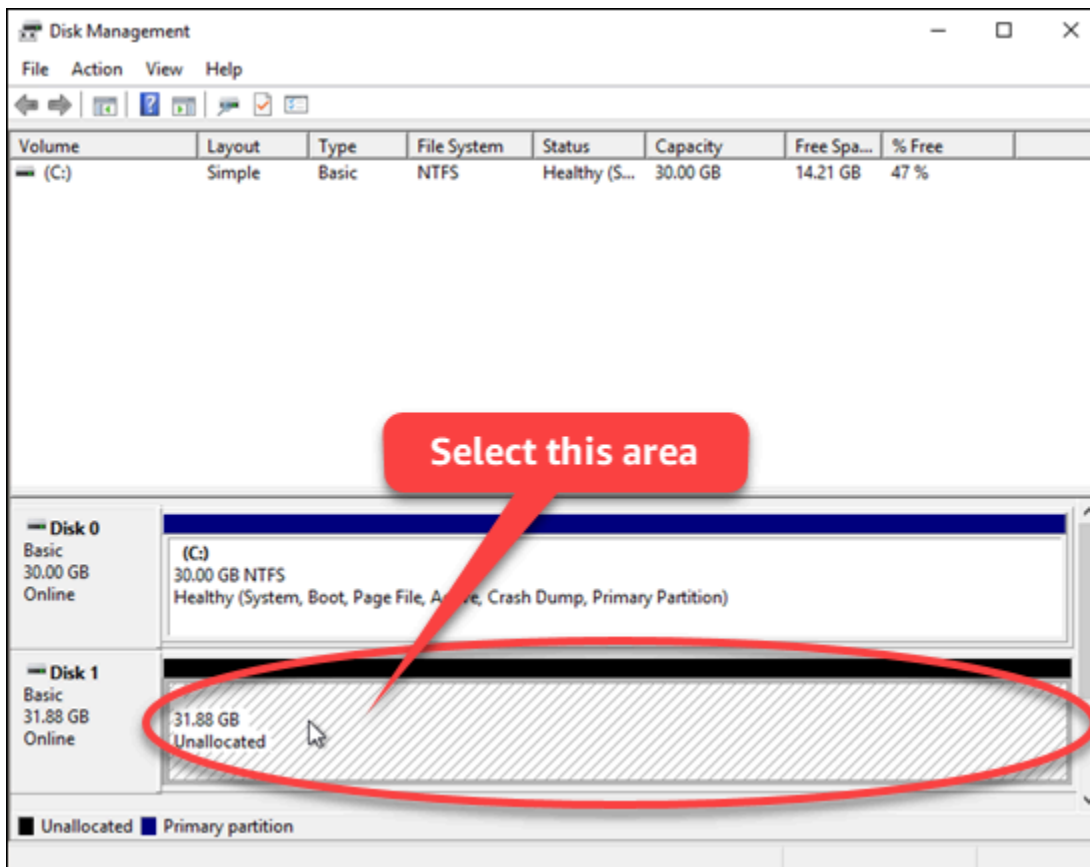
You should see the status of the block storage disk update to **Online**. Continue to the [Step 4: Format the disk with a file system](#) section of this guide to format your block storage disk with a file system.

Step 4: Format the disk with a file system

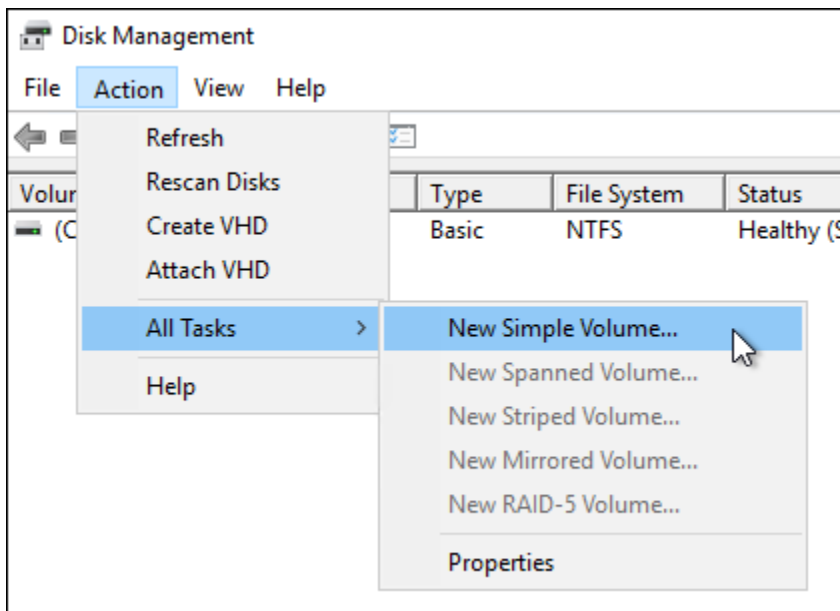
Use the New Simple Volume wizard in Windows Server to assign a drive letter and format the disk with a file system.

To format the disk with a file system

1. In the bottom pane of the Disk Management utility, select the partition on the block storage disk labeled as **Unallocated**.



2. With the partition selected, on the **Action** menu, choose **All Tasks**, and then choose **New Simple Volume**.

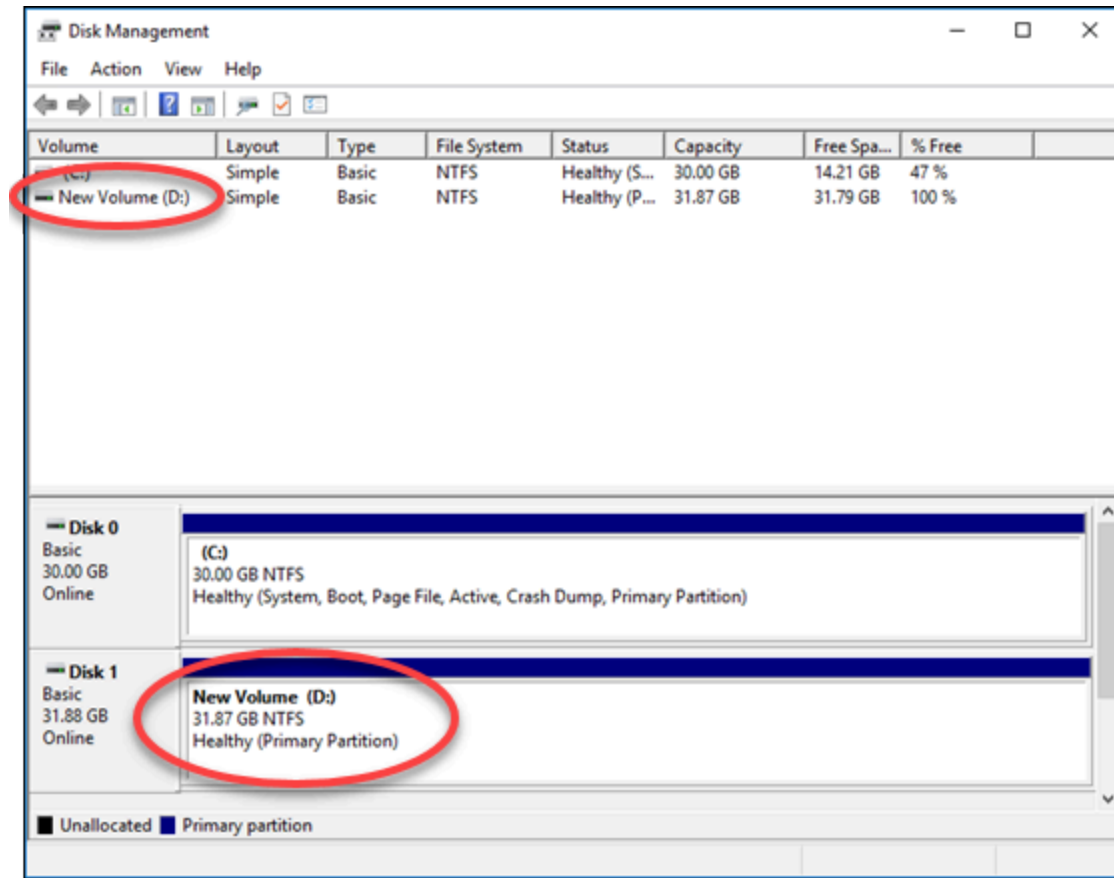


3. Follow the instructions in the New Simple Volume wizard to choose an NTFS, FAT32, or ReFS file system type and format the disk.

Note

For more information about each of these file systems, see the [NTFS overview](#), [Resilient File System \(ReFS\) overview](#), and [Description of the FAT32 File System](#) articles from Microsoft.

When complete, you see a drive letter and the following message in the Disk Management utility.



Detach and delete a Lightsail block storage disk

If you no longer need a block storage disk, you can detach it from your stopped Lightsail instance, and then delete it. This topic describes how to back up your data and safely delete a disk.

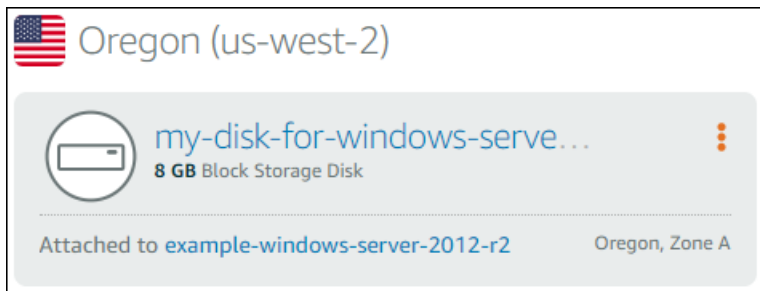
Prerequisites

- Stop your instance from running. You have to do this before you can detach and then delete your disk. [Learn how to stop your instance](#)
- (Optional) We recommend that you create a snapshot of your disk. That way, you have a backup in case you change your mind. For more information, see [Create a snapshot of your database](#)

Detach and delete your disk

Once you stop your Lightsail instance, you can safely detach and delete your disk.

1. On the home page, choose **Storage**.
2. Choose the name of your attached disk to manage it.



3. On the disk management page, choose **Detach**.

After a few seconds, the disk is detached and ready to be deleted or reattached.

4. Choose the **Delete** tab.
5. Choose **Delete disk**, and then confirm by choosing **Yes, delete**.

Important

This is a permanent operation and can't be undone. You will lose all data on the disk when you delete it.

Snapshots in Amazon Lightsail

You can create point-in-time snapshots of instances, databases, and block storage disks in Amazon Lightsail, and use them as baselines to create new resources or for data backup. A snapshot contains all of the data that is needed to restore your resource (from the moment when the snapshot was taken). When you restore a resource by creating it from a snapshot, the new resource begins as an exact replica of the original resource that was used to create the snapshot. You will be billed a [snapshot storage fee](#) for snapshots on your Lightsail account; whether they are manual snapshots, automatic snapshots, copied snapshots, or system disk snapshots. If you experience data corruption or a disk failure, you can create a disk from a snapshot that you have taken and replace the old disk. You can also use snapshots to provision new disks and attach them during a new instance launch.

Contents

- [Manual snapshots](#)
- [Automatic snapshots](#)
- [System disk snapshots](#)
- [Create new resources from snapshots](#)
- [Copy snapshots](#)
- [Export snapshots to Amazon EC2](#)
- [Delete snapshots](#)

Manual snapshots

Create manual snapshots of instances, managed databases, and block storage disks at any time. Manual snapshots are stored indefinitely until you delete them.

For more information about creating manual snapshots, see the following guides:

- [Create a snapshot of your Linux or Unix instance](#)
- [Create a snapshot of your Windows Server instance](#)
- [Create a snapshot of your database](#)
- [Create a block storage disk snapshot](#)

Automatic snapshots

If you're hosting critical information on your Lightsail instance or block storage disk, you should back them up often by creating manual snapshots. However, it's not always easy to find the time to perform frequent administrative tasks. If that's the case for you, then use automatic snapshots to have Lightsail create daily backups of your instance or block storage disk on your behalf, without manual interaction. The latest seven daily automatic snapshots are stored before the oldest one is replaced with the newest one.

For more information about automatic snapshots, see the following guides:

- [Enable or disable automatic instance snapshots](#)
- [Change the automatic snapshot time for instances or disks](#)
- [Delete automatic snapshots](#)

Important

All automatic snapshots associated with a resource are deleted when you delete the source resource. This behavior differs from manual snapshots, which are kept in your Lightsail account even after you delete the source resource. To keep your automatic snapshots when you delete the source resource, see [Keep automatic snapshots](#).

System disk snapshots

If your instance becomes unresponsive and you need to access the files on the system disk, you can back up the instance root volume by creating a snapshot of it. Then, you can access the files in the system disk by creating a new block storage disk from the snapshot and attaching it to another instance. For more information, see [Create a snapshot of an instance root volume](#).

Create new resources from snapshots

Use snapshots to create new Lightsail resources using the same plan, or larger plan, than the original resource. When you create a resource based on a snapshot, the new resource begins as a replica of the original resource that was used to create the snapshot. Snapshots cannot be used to create new resources using a smaller Lightsail plan.

For more information, see the following guides:

- [Create an instance from a snapshot](#)
- [Create a database from a snapshot](#)
- [Create a block storage disk from a snapshot](#)
- [Create a larger instance, block storage disk, or database from a snapshot](#)

Copy snapshots

Instance and block storage disk snapshots can be copied from one Amazon Web Services (AWS) Region to another within the same Lightsail account. Database snapshots cannot be copied between regions. For more information, see [Copy snapshots from one AWS Region to another](#).

Export snapshots to Amazon EC2

Lightsail is the easiest way to get started with AWS. However, there are limitations with Lightsail that are not present in Amazon EC2 or other AWS services. Export your Lightsail instance and block storage disk snapshots to Amazon EC2 to take advantage of the wider range of instance types available, and use the full range of services in AWS. For more information, see [Export snapshots to Amazon EC2](#).

Note

Snapshots of cPanel and WHM, Django, and Ghost instances cannot be exported to Amazon EC2 at this time.

Delete snapshots

Delete Lightsail snapshots when you no longer need them to avoid incurring a monthly [snapshot storage fee](#). For more information, see [Delete snapshots](#).

Create a Lightsail block storage disk snapshot

You can create disk snapshots in Lightsail as backups of your additional block storage disks.

You can use the snapshot of a disk as a baseline for new disks or for data backup. If you make periodic snapshots of a disk, the snapshots are incremental. Only the blocks on the device that have changed after your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot to restore the entire disk.

For more information, see [Snapshots](#).

1. On the Lightsail home page, choose the **Storage** tab.
2. Choose the name of the block storage disk for which you want to create a snapshot.
3. Choose the **Snapshots** tab.
4. Under the **Manual snapshots** section of the page, choose **Create snapshot**, then enter a name for your snapshot.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
5. Choose **Create**.

You can see the snapshot you just created with a status of **Snapshotting...**

After the snapshot is finished, you can [create another disk from the snapshot](#).

Create a Lightsail block storage disk from a snapshot

You can create a new block storage disk from a disk snapshot. If you're creating an entirely new disk, see one of the following topics instead: [Create additional block storage disks \(Linux/Unix\)](#) or [Create and attach block storage disks to your Windows Server instance](#).

You can use the snapshot of a block storage disk as a baseline for new disks or for data backup. If you make periodic snapshots of a disk, the snapshots are incremental. Only the blocks on the disk that have changed after your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only

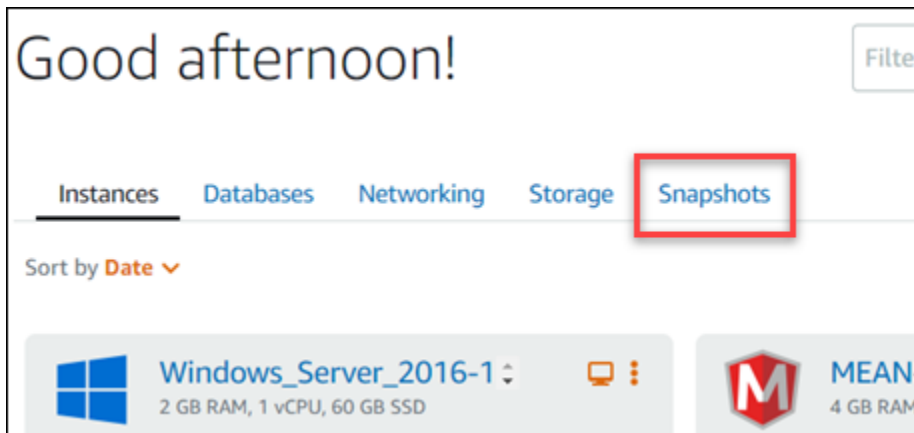
the most recent snapshot to restore the entire disk. To create a snapshot of your block storage disk, see [Create a block storage disk snapshot](#).

Step 1: Find your disk snapshot and choose to create a new disk

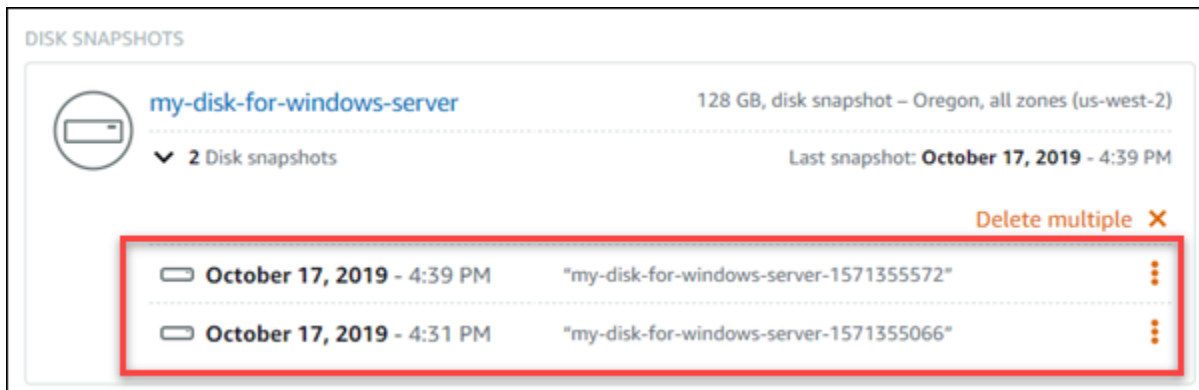
You can create a new instance from a disk snapshot in one of two places in Lightsail: on the **Snapshots** tab of the Lightsail home page, or on the **Snapshots** tab of the disk management page.

From the Lightsail home page

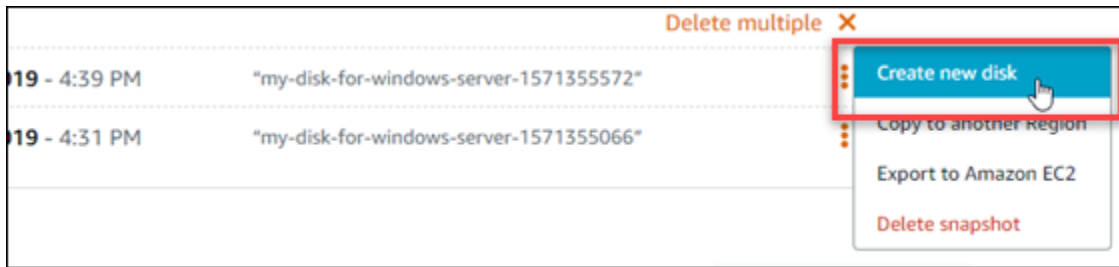
1. On the Lightsail home page, choose the **Snapshots** tab.



2. Find the name of the disk, then expand the node below it to see all of the available snapshots of that disk.

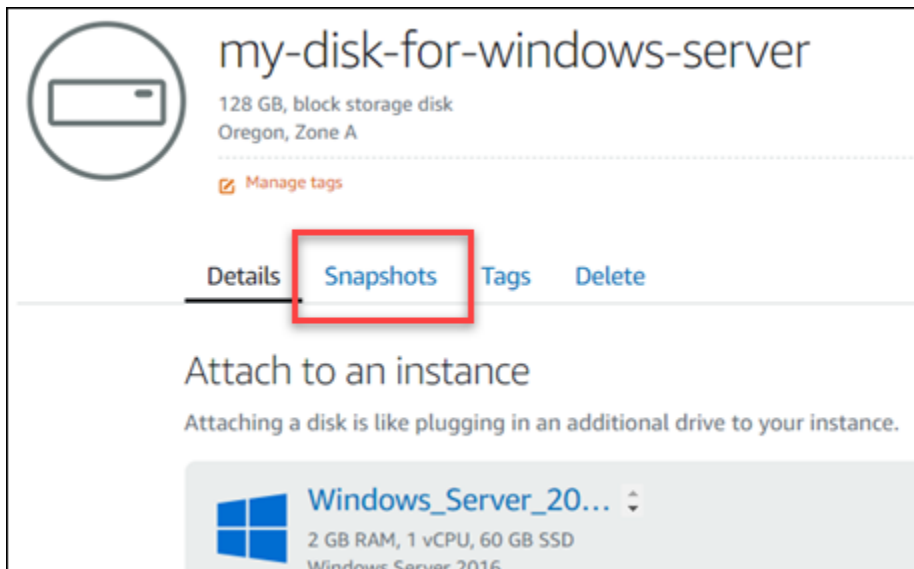


3. Choose the actions menu icon (:) next to the snapshot from which you want to create your new disk, and then choose **Create new disk**.

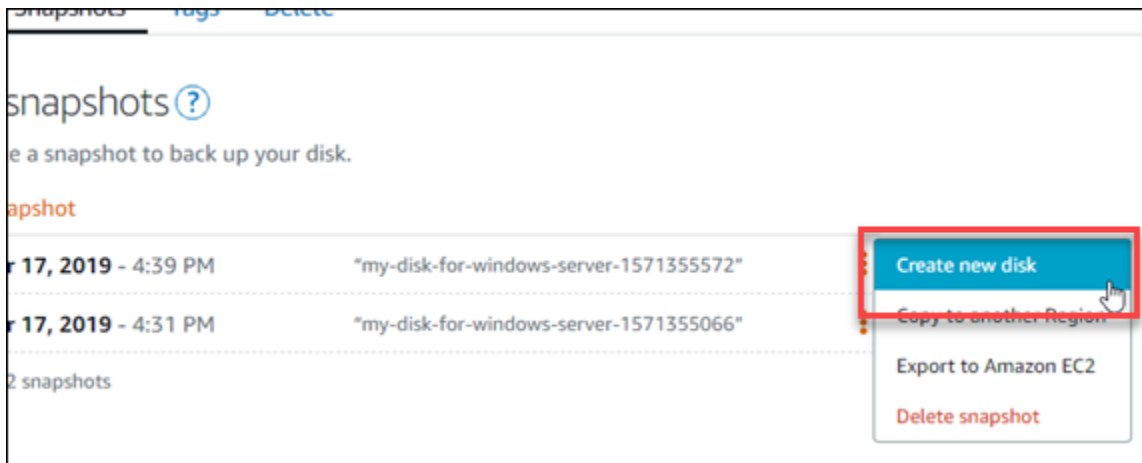


From the disk management page in Lightsail

1. On the Lightsail home page, choose the **Storage** tab.
2. Choose the name of the disk for which you want to view snapshots.
3. Choose the **Snapshots** tab.



4. Under the **Manual snapshots** section of the page, choose the actions menu icon (:) next to the snapshot from which you want to create a new disk, and choose **Create new disk**.



Step 2: Create a new disk from a disk snapshot

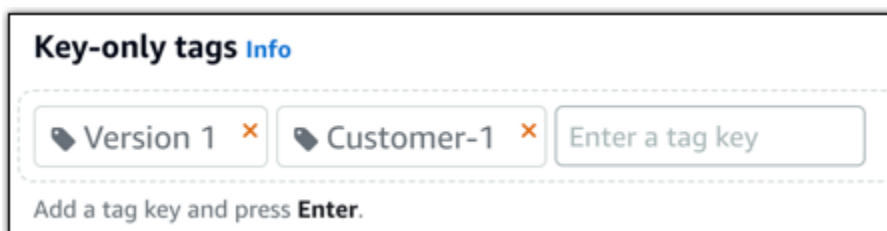
1. Choose an Availability Zone for your new disk, or accept the default (e.g., us-east-2a).

You must create the new disk in the same AWS Region as the source disk.

2. Choose a size for your new disk that is equal to or greater than the source snapshot.
3. Enter a name for your disk.

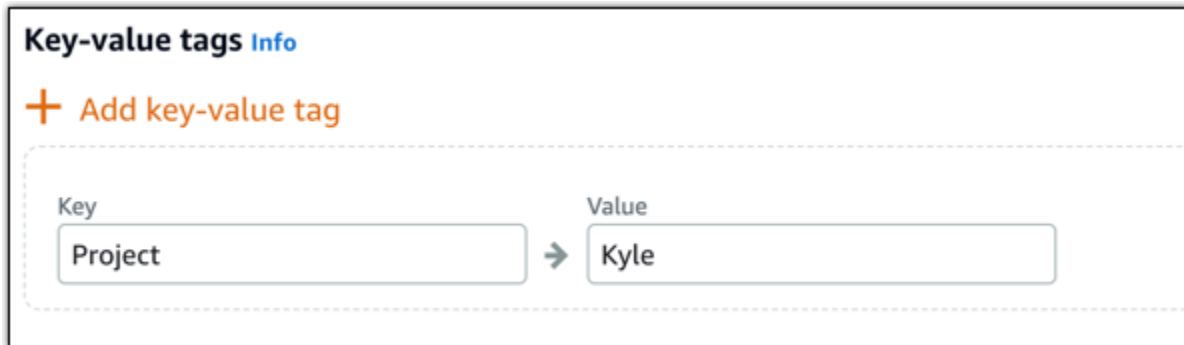
Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
4. Choose one of the following options to add tags to your disk:
 - **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



The screenshot shows a user interface for adding key-value tags. At the top, it says "Key-value tags Info". Below that is a button with a plus sign and the text "Add key-value tag". A dashed-line box contains two input fields: "Key" with the value "Project" and "Value" with the value "Kyle", connected by a right-pointing arrow.

Note

For more information about key-only and key-value tags, see [Tags](#).

5. Choose **Create disk**.

Create a snapshot of a Lightsail instance's root volume

Back up an instance root volume in Amazon Lightsail by creating a snapshot of the system disk. Then, access the files in the backup by creating a new block storage disk from the snapshot and attaching it to another instance. Do this if you need to:

- Recover data from the root volume of a botched instance.
- Create a backup of your instance's root volume, as you would for a block storage disk.

You create the instance root volume snapshot using the AWS Command Line Interface (AWS CLI). After you create the snapshot, use the Lightsail console to create a block storage disk from the snapshot. Then, attach it to a running instance, and access it from that instance.

Contents

- [Step 1: Complete the prerequisites](#)

- [Step 2: Create an instance root volume snapshot](#)
- [Step 3: Create a block storage disk from a snapshot and attach it to an instance](#)
- [Step 4: Access a block storage disk from an instance](#)

Step 1: Complete the prerequisites

If you haven't done so already, install and configure the AWS CLI. For more information, see [Configure the AWS CLI to work with Lightsail](#).

Step 2: Create an instance root volume snapshot

Open a Terminal or Command Prompt window, then type the following command to create an instance root volume snapshot.

```
aws lightsail create-disk-snapshot --region AWSRegion --instance-name InstanceName --disk-snapshot-name DiskSnapshotName
```

In the command, replace:

- *AWSRegion* with the AWS Region of the instance.
- *InstanceName* with the name of the instance whose root volume you want to back up.
- *DiskSnapshotName* with the name of the new disk snapshot to be created.

Example:

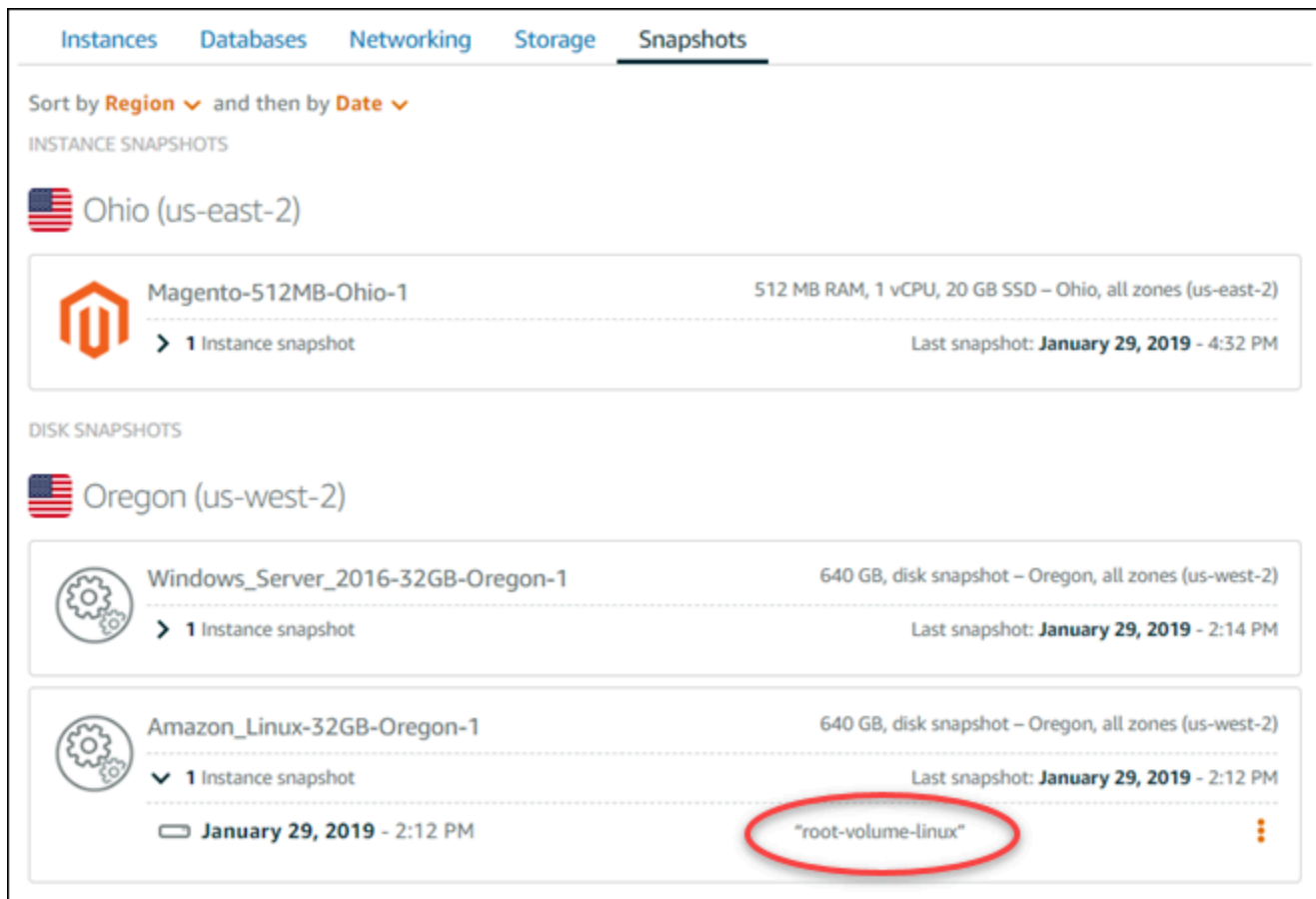
```
aws lightsail create-disk-snapshot --region us-west-2 --instance-name Amazon_Linux-32MB-Oregon-1 --disk-snapshot-name root-volume-linux
```

If successful, you will see a result similar to the following:

```
H:\>aws lightsail create-disk-snapshot --region us-west-2 --instance-name Amazon_Linux-32GB-Oregon-1
--disk-snapshot-name root-volume-linux

{
  "operations": [
    {
      "status": "Started",
      "resourceType": "DiskSnapshot",
      "isTerminal": false,
      "operationDetails": "Amazon_Linux-32GB-Oregon-1",
      "statusChangedAt": 1548799955.599,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "operationType": "CreateDiskSnapshot",
      "resourceName": "root-volume-linux",
      "id": "disk-snapshot-arn:aws:lightsail:us-west-2:123456789012:disk-snapshot-123456789012",
      "createdAt": 1548799955.599
    },
    {
      "status": "Started",
      "resourceType": "Instance",
      "isTerminal": false,
      "operationDetails": "root-volume-linux",
      "statusChangedAt": 1548799955.599,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "operationType": "CreateDiskSnapshot",
      "resourceName": "Amazon Linux-32GB-Oregon-1",
      "id": "instance-arn:aws:lightsail:us-west-2:123456789012:instance-123456789012",
      "createdAt": 1548799955.599
    }
  ]
}
```

Wait a few minutes for the snapshot to be created. After it's created, you can view it in the Lightsail home page by choosing the **Snapshots** tab and scrolling to the Disk Snapshots section, as shown in the following example.



The screenshot shows the AWS Lightsail Snapshots console. At the top, there are navigation tabs for Instances, Databases, Networking, Storage, and Snapshots. Below the tabs, there are sorting options: 'Sort by Region' and 'and then by Date'. The console is divided into two sections: 'INSTANCE SNAPSHOTS' and 'DISK SNAPSHOTS'. Under 'INSTANCE SNAPSHOTS', there is a section for 'Ohio (us-east-2)' which contains one instance snapshot named 'Magento-512MB-Ohio-1'. Under 'DISK SNAPSHOTS', there are two sections: 'Oregon (us-west-2)'. The first section contains a disk snapshot named 'Windows_Server_2016-32GB-Oregon-1'. The second section contains a disk snapshot named 'Amazon_Linux-32GB-Oregon-1'. Within this second section, there is a sub-section for the snapshot 'January 29, 2019 - 2:12 PM' which contains a single snapshot named 'root-volume-linux'. This snapshot name is circled in red in the original image.

Step 3: Create a block storage disk from a snapshot and attach it to an instance

Create a new block storage disk from the instance root volume snapshot and attach it to another instance if you must access its contents. Do this if you need to recover data from the root volume of a botched instance.

Note

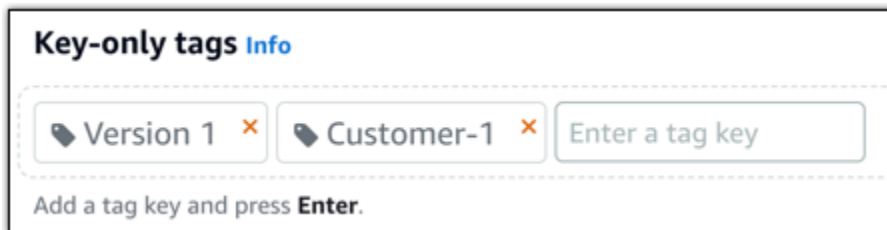
The new block storage disk is created in the same AWS Region as the source snapshot. To create the block storage disk in a different Region, copy the snapshot to the desired Region, and then create a new disk from the copied snapshot. For more information, see [Copy snapshots from one AWS Region to another](#).

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Snapshots** tab.

3. Choose the actions menu icon (:) displayed next to the root volume disk snapshot that you want to use, then choose **Create new disk**.
4. Choose an Availability Zone for the disk, or accept the default.
5. Choose a size for the disk that is equal to or greater than the source disk.
6. Enter a name for the disk.

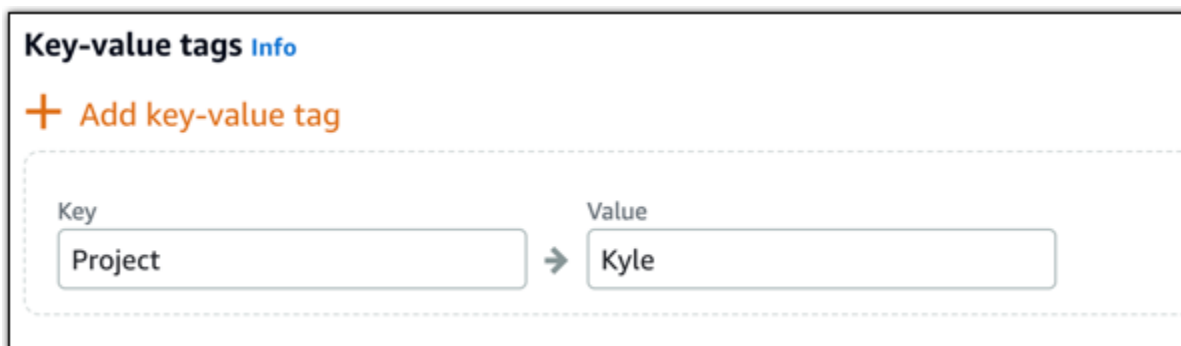
Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
7. Choose one of the following options to add tags to your disk:
 - **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

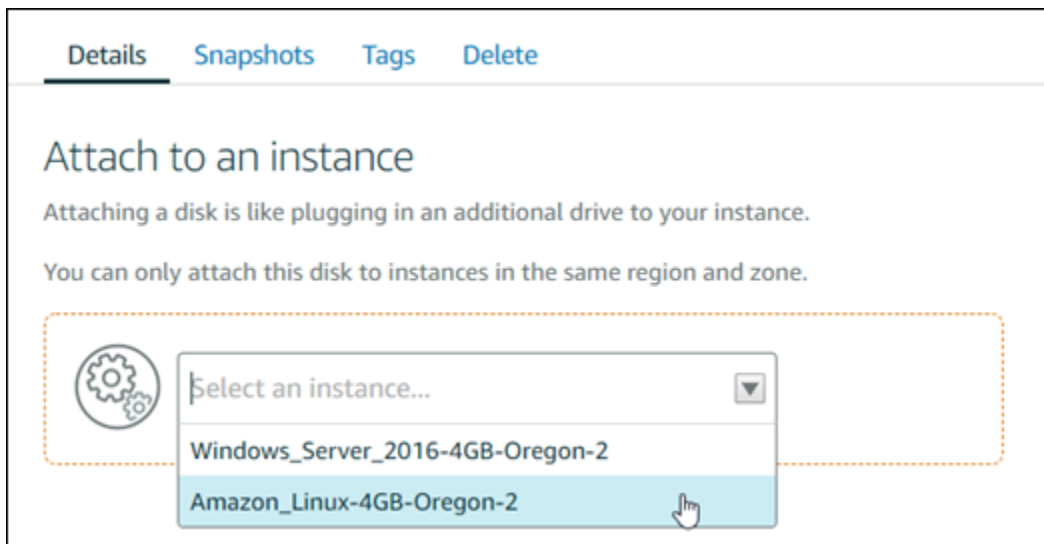
Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Note

For more information about key-only and key-value tags, see [Tags](#).

8. Choose **Create disk**.
9. After the disk is created, choose the instance that you want to attach the disk to in the **Select an instance** drop-down menu. This is shown in the following example.



10. Choose **Attach** to attach the disk to the selected instance.

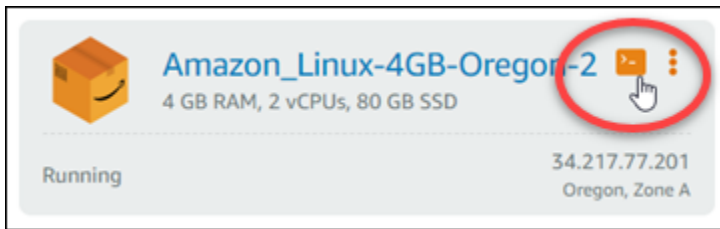
The disk is now attached to the instance. Next, make it accessible to the applicable operating system by mounting it on Linux, or bringing it online on Windows. For more information, see the following **Access the block storage from an instance** section of this guide.

Step 4: Access a block storage disk from an instance

To access a block storage disk after attaching it to an instance, you must mount it on Linux or Unix, or bring it online on Windows.

Mount and access a block storage disk on a Linux or Unix instance

1. On the [Lightsail home page](#), choose the browser-based SSH client icon for the Linux or Unix instance to which you attached the block storage disk.



2. After the browser-based SSH client is connected, enter the following command to view the block storage disk devices attached to the instance:

```
lsblk
```

You should see a result similar to the following example. In this example, `xvdf1` is the block storage disk attached to the instance that is not yet mounted because it doesn't have a mount point. Also, the result omits `/dev/` from the device name, so the device name is actually `/dev/xvdf1`.

```
[ec2-user@ip-172-31-0-111 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   80G  0  disk
└─xvda1     202:1    0   80G  0  part /
xvdf        202:80   0  640G  0  disk
└─xvdf1     202:81   0  640G  0  part
```

3. Enter the following command to create a mount point for the block storage disk.

```
sudo mkdir MountPoint
```

In the command, replace *MountPoint* with the name of the directory where the block storage disk will be mounted and accessible.

Example:

```
sudo mkdir xvdf
```

4. Enter the following command to mount the block storage disk to the mount point you created in the previous step.

```
sudo mount /dev/DeviceName MountPoint
```

In the command, replace:

- *DeviceName* with the name of the block storage disk device.
- *MountPoint* with the mount point directory that you created in the previous step.

Example:

```
sudo mount /dev/xvdf1 xvdf
```

5. Enter the following command to view the block storage disk devices attached to the instance:

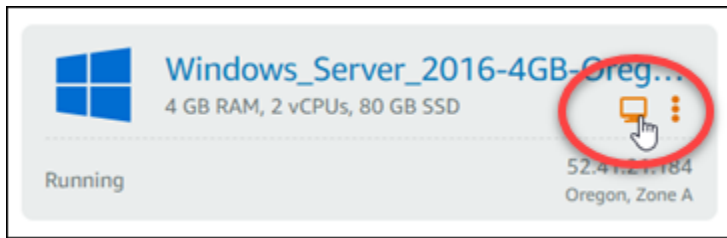
```
lsblk
```

You should see a result similar to the following example. In this example, the *xvdf1* device is now mounted and accessible at the */home/ec2-user/xvdf* directory. You can now access block storage disk and its contents by going to the mount point directory.

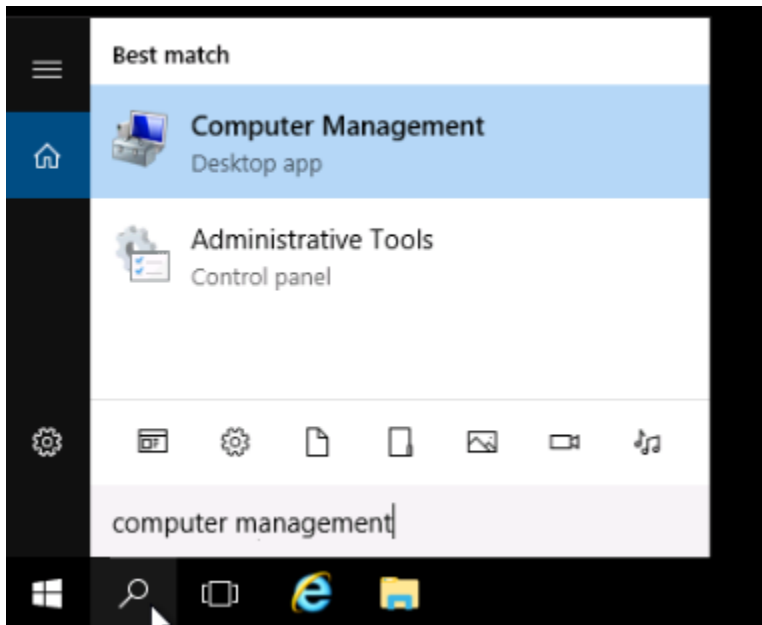
```
[ec2-user@ip-10-0-1-10 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   80G  0 disk
└─xvda1     202:1    0   80G  0 part /
xvdf        202:80   0  640G  0 disk
└─xvdf1     202:81   0  640G  0 part /home/ec2-user/xvdf
```

Bring a block storage disk online and access it on a Windows instance

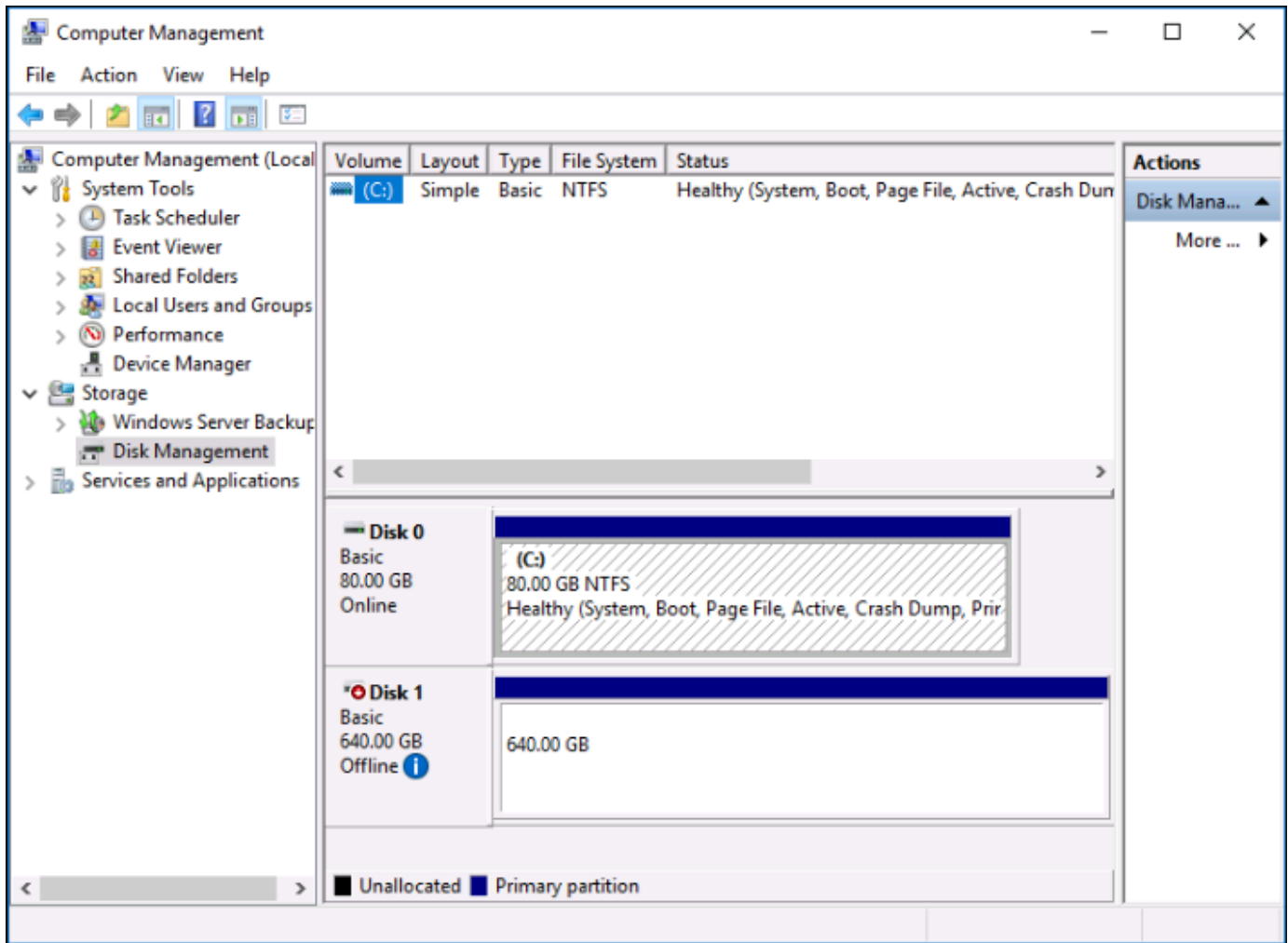
1. On the [Lightsail home page](#), choose the browser-based RDP client icon for the Windows instance to which you attached the block storage disk.



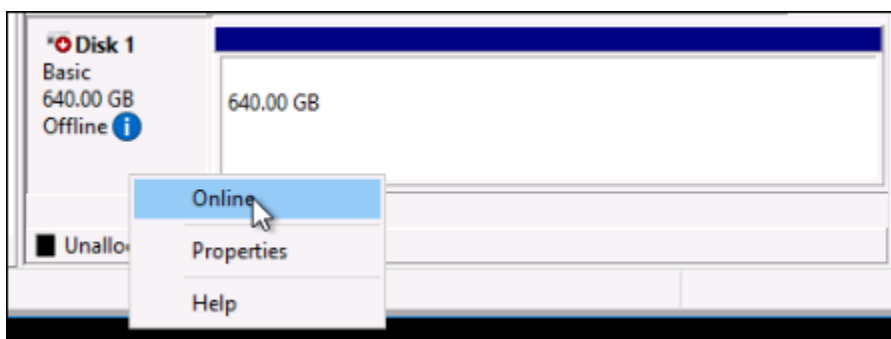
2. After the browser-based SSH client is connected, search for **Computer Management** in the Windows taskbar, then choose **Computer Management** from the results.



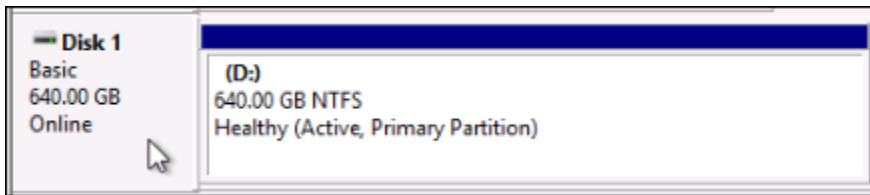
3. In the left navigation menu of the **Computer Management** console, choose **Disk Management**, as shown in the following example.



4. Locate the disk that you recently attached to the instance. It should be labeled as **Offline**.
5. Right-click the **Offline** label, then choose **Online**.



The disk should now be labeled as **Online**, and a drive letter should be associated with it. You can now access the block storage disk and its contents by opening File Explorer and browsing to the designated drive letter.

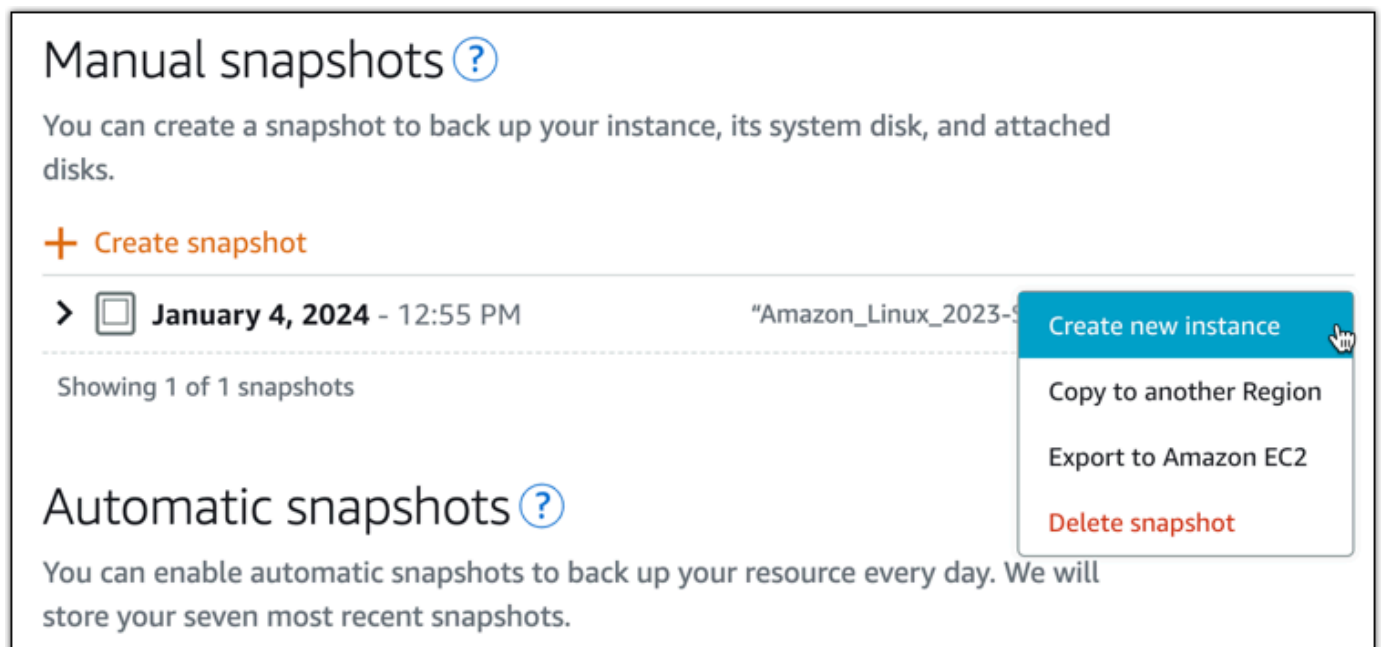


Create a Lightsail instance from a snapshot


After you create a snapshot in Lightsail, you can create a new instance from that snapshot. You can change attributes of the new instance, such as instance size and networking type—dual-stack or IPv6-only. The new instance includes the system disk and the attached block storage disks that you added.

You must have a snapshot of an instance before you can create another instance from that snapshot. For more information, see [Create a snapshot of your Linux or Unix Lightsail instance](#) or [Create a snapshot of your Lightsail Windows Server instance](#).

1. On the Lightsail console, choose the instance that you want to snapshot to create a new instance.
2. Choose the **Snapshots** tab.
3. In the **Manual snapshots** section, choose the actions menu icon (:) next to the snapshot and choose **Create new instance**.



- The **Create an instance from a snapshot** page opens. Choose the optional settings that you want to use. For example, you can change the Availability Zone, [add a launch script](#), or [change the way you connect to your instance](#).
- Choose a plan (or *bundle*) for your new instance. You can choose to create an instance that uses a dual-stack (IPv4 and IPv6) instance plan, or an IPv6-only plan. You can also choose a larger bundle size than that of the original instance. For more information about IPv6-only instance plans, see [IPv6-only instance plans in Lightsail](#).

 **Note**

You can't create an instance that uses a smaller bundle size than that of the original instance.

Choose a new instance plan Info

You can pick a machine the same size or larger than the source snapshot.



Select an IP address type - *new* Info

Dual stack Recommended

Includes both a public IPv4 and IPv6 address. Suitable for most use cases due to wide compatibility with IPv4 addresses.

IPv6 only

Includes a public IPv6 address. An advanced option for use cases where IPv6 access limitations are acceptable.

 **Updated pricing for instances with public IPv4** Learn more 

Starting June 1, 2024, all Lightsail instance bundles that include a public IPv4 address will incur a new price. You can now launch IPv6-only bundles if your instance doesn't require a public IPv4 address.

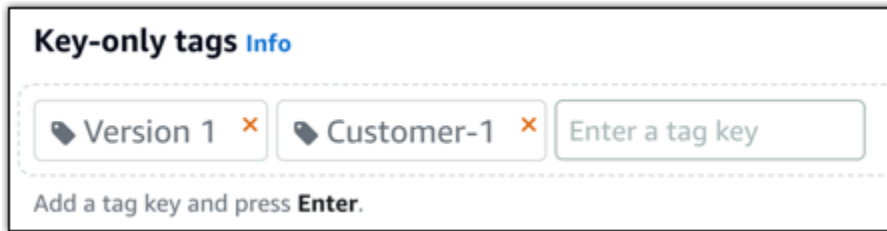
- Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region of your Lightsail account.
- Must contain 2–255 characters.
- Must start and end with an alphanumeric character.
- Can include alphanumeric characters, periods, dashes, and underscores.

- Choose one of the following options to add tags to your instance:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the text box, and press **Enter**. Choose **Save** or **Cancel**.



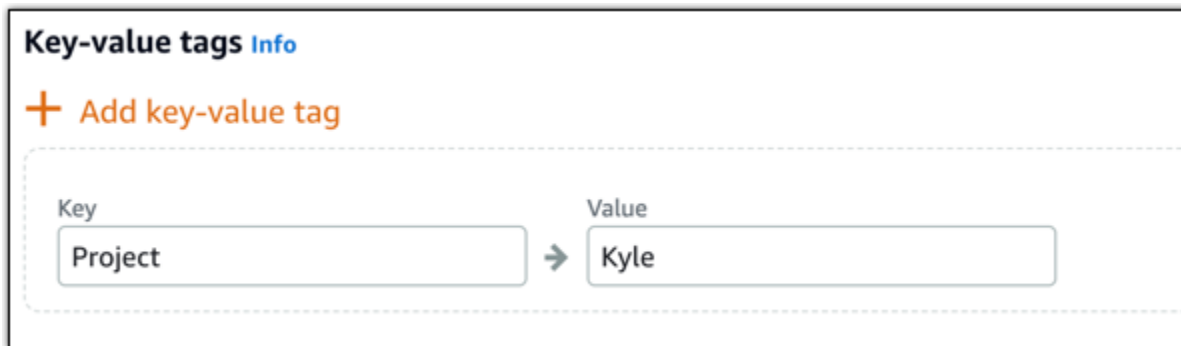
Key-only tags Info

Version 1 × Customer-1 × Enter a tag key

Add a tag key and press **Enter**.

- **Create a key-value tag**, then enter a key into the **Key** text box and a value into the **Value** text box. Choose **Save** or **Cancel**.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key Value

Project → Kyle

Note

For more information about key-only and key-value tags, see [Tags](#).

8. Choose **Create instance**.

Lightsail opens the management page, where you can manage your new instance.

Important

Custom firewall rules from the original instance don't copy over to the new instance that you create from a snapshot. Only the default rules copy over to the new instance. For more information, see [Default instance firewall rules](#).

Create a larger instance, block storage disk, or database from a Lightsail snapshot

It happens. Your cloud project is growing and you need more compute power right away! We can help you with that. To upsize your Lightsail instance, block storage disk, or database, create a snapshot of your resource, and then create a new, larger version of that resource using the snapshot.

Note

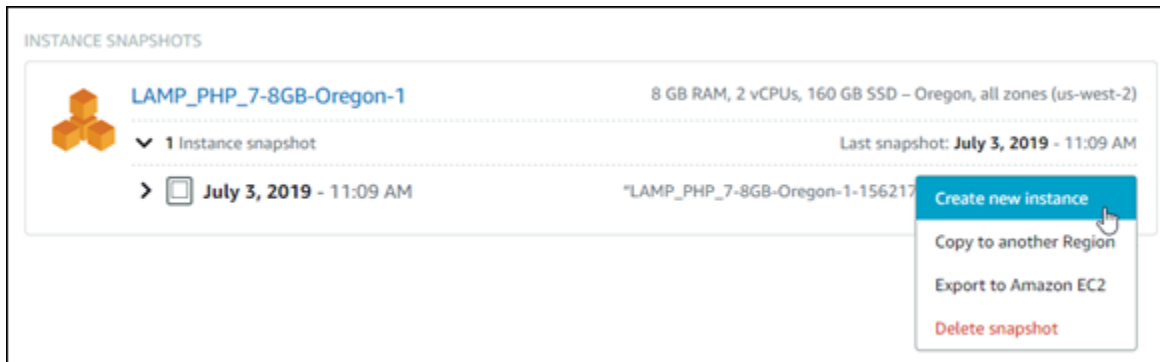
You cannot create a resource from a snapshot using a smaller plan size than the original resource. For example, you can't go from an 8 GB instance to a 2 GB instance. The default public IPv4 address that is assigned to your instance when you create it will change when you stop and start your instance. You can optionally create and attach a static IPv4 address to your instance. By using a static IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Alternatively, you can specify the static IP address in a DNS record for your domain, so that your domain points to your instance. For more information, see [IP addresses](#).

Prerequisites

You'll need a snapshot of your Lightsail instance, block storage disk, or database. For more information, see [Snapshots](#).

Create your resource

1. Sign in to the [Lightsail console](#).
2. Choose the **Snapshots** tab.
3. Find the Lightsail resource whose snapshot you want to use to create a new, larger resource, and choose the right-arrow to expand the list of snapshots.
4. Choose the ellipsis icon next to the snapshot you want to use, and choose **Create new**.



5. On the **Create** page, you have a few optional settings to choose from. For example, you can change the Availability Zone. For instances, you can [add a launch script](#), or [change the SSH key you use to connect to it](#).

You can accept all the defaults and move on to the next step.

6. Choose the plan (or *bundle*) for your new resource. At this point, you can choose a larger bundle size than the original resource, if you'd like.

Note

You cannot create the resource using a smaller plan size than the original resource. The bundle options that are smaller than the original resource will be unavailable.

7. Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

8. Choose **Create**.

Lightsail takes you to the management page for your new resource, and you can start managing it.

Create a larger instance, block storage disk, or database from a Lightsail snapshot using the AWS CLI

It happens. Your cloud project is growing and you need more compute power right away! We can help you with that. You can do everything from within the Lightsail console, or you can use the AWS Command Line Interface (AWS CLI) to do it.

We'll show you how to take a *snapshot* of your current Lightsail instance and create a new, larger one with the compute power you need based on that snapshot.

Note

At this time, we don't support creating a smaller instance size (or bundle) from a snapshot. You can only create the same size instance or a larger instance.

Prerequisites

1. First, if you haven't already, you need to install the AWS CLI. To learn more, see [Installing the AWS Command Line Interface](#). Be sure [you configure the AWS CLI](#).
2. You also need a snapshot of your instance to work from. To learn more, see [Create a snapshot of your Linux or Unix instance](#).

Step 1: Get your snapshot name

This might seem obvious, but you need to have your snapshot name before you execute this AWS CLI command to create the larger instance. The good news is that it's easy to get.

1. In the AWS CLI, type the following.

```
aws lightsail get-instance-snapshots
```

You should see output similar to the following.

```
{
  "instanceSnapshots": [
    {
      "fromInstanceName": "WordPress-512MB-EXAMPLE",
```

```
    "name": "WordPress-512MB-EXAMPLE-system-1234567891011",
    "sizeInGb": 20,
    "resourceType": "InstanceSnapshot",
    "fromInstanceArn":
    "arn:aws:lightsail:us-
east-1:123456789101:Instance/86f49ee4-26cc-4802-9b0d-12345EXAMPLE",
    "state": "available",
    "arn": "arn:aws:lightsail:us-east-1:123456789101:InstanceSnapshot/
c87acb5f-851e-4fbc-94f1-12345EXAMPLE",
    "fromBundleId": "nano_1_0",
    "fromBlueprintId": "wordpress_4_6_1",
    "createdAt": 1480898073.653,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-east-2"
    }
  }
]
```

2. Copy the **name** value to some place where you can get it later. This is the `--instance-snapshot-name` value you'll use in your AWS CLI command.

Step 2: Choose a bundle

A *bundle* is really just a pricing plan and a configuration for your instance. For example, **Medium** Linux-based bundles cost \$20 USD per month and have 4.0 GB of RAM, 80 GB SSD storage, and so on.

If you started out with a smaller bundle and need more compute power, you might want to upgrade to a larger bundle. For more information, see [Create a larger instance, block storage disk, or database from a snapshot](#).

Important

You cannot resize to a smaller bundle from a snapshot. If you want to create a smaller bundle, you have to start over.

1. Type the following AWS CLI command.

```
aws lightsail get-bundles
```

Your output should be similar to the following.

```
{
  "bundles": [
    {
      "name": "Nano",
      "power": 300,
      "price": 5.0,
      "ramSizeInGb": 0.5,
      "diskSizeInGb": 20,
      "transferPerMonthInGb": 1024,
      "cpuCount": 1,
      "instanceType": "t2.nano",
      "isActive": true,
      "bundleId": "nano_1_0"
    },
    {
      "name": "Micro",
      "power": 500,
      "price": 10.0,
      "ramSizeInGb": 1.0,
      "diskSizeInGb": 30,
      "transferPerMonthInGb": 2048,
      "cpuCount": 1,
      "instanceType": "t2.micro",
      "isActive": true,
      "bundleId": "micro_1_0"
    },
    {
      "name": "Small",
      "power": 1000,
      "price": 20.0,
      "ramSizeInGb": 2.0,
      "diskSizeInGb": 40,
      "transferPerMonthInGb": 3072,
      "cpuCount": 1,
      "instanceType": "t2.small",
      "isActive": true,
      "bundleId": "small_1_0"
    },
  ],
}
```

```
{
  "name": "Medium",
  "power": 2000,
  "price": 40.0,
  "ramSizeInGb": 4.0,
  "diskSizeInGb": 60,
  "transferPerMonthInGb": 4096,
  "cpuCount": 2,
  "instanceType": "t2.medium",
  "isActive": true,
  "bundleId": "medium_1_0"
},
{
  "name": "Large",
  "power": 3000,
  "price": 80.0,
  "ramSizeInGb": 8.0,
  "diskSizeInGb": 80,
  "transferPerMonthInGb": 5120,
  "cpuCount": 2,
  "instanceType": "t2.large",
  "isActive": true,
  "bundleId": "large_1_0"
}
]
```

2. Locate the **bundleId** value of the bundle you want. For more information, see [Lightsail Pricing](#).

Step 3: Write your AWS CLI command and create your new instance

Now that you have your parameter values, you're ready to write and execute your command to create the instance!

1. Type the following.

```
aws lightsail create-instances-from-snapshot --instance-names
MyNewInstanceFromSnapshot --availability-zone us-east-1a --instance-snapshot-name
WordPress-512MB-EXAMPLE-system-1234567891011 --bundle-id medium_1_0
```

Your output should be similar to the following.

```
{
  "operations": [
    {
      "status": "Started",
      "resourceType": "Instance",
      "isTerminal": false,
      "statusChangedAt": 1486863990.961,
      "location": {
        "availabilityZone": "us-east-2a",
        "regionName": "us-east-2"
      },
      "operationType": "CreateInstance",
      "resourceName": "MyNewInstanceFromSnapshot",
      "id": "30fec45e-e7d7-4e18-96c8-12345EXAMPLE",
      "createdAt": 1486863989.784
    }
  ]
}
```

Note

You can also return a list of regions and Availability Zones using the AWS CLI. Just type `aws lightsail get-regions --include-availability-zones` to return the list of availability zones with your `get-regions` request.

2. Now open your new instance in the Lightsail console and start modifying it.

Next steps

After you create your new instance from a snapshot, here are some things you can do next:

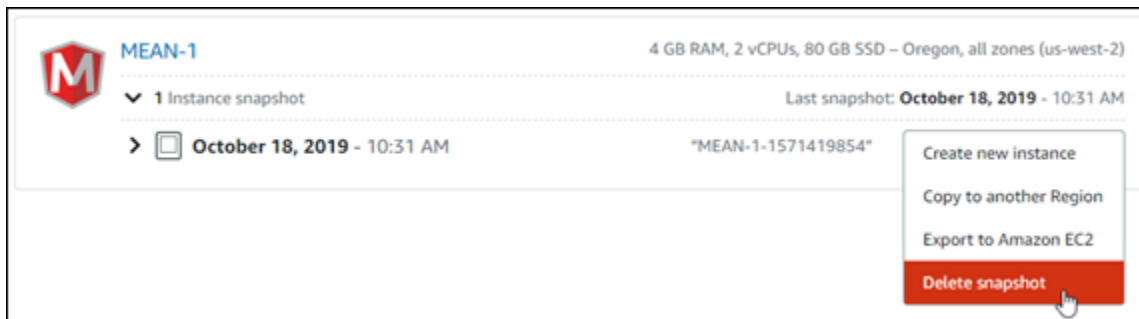
- If you're done with the old instance, you might want to delete it. You can do this by using the Lightsail console or the [delete-instance CLI command](#).
- If you don't need the old snapshot, you might want to delete it. You can do this by using the Lightsail console or the [delete-instance-snapshot CLI command](#).
- If you had a static IP address attached to your old instance, you might want to keep it and attach it to the new instance. You can do this by using the console. See [Create a static IP and attach it to an instance](#).

Delete Lightsail snapshots

Delete instance, database, and disk snapshots in Amazon Lightsail if you no longer need them to avoid incurring a monthly charge.

Delete an individual snapshot

1. On the [Lightsail console](#), choose **Snapshots** tab.
2. Find the Lightsail resource whose snapshot you want to delete, and choose the right-arrow to expand the list of available snapshots for that resource.
3. Choose the actions menu icon (:) next to the snapshot you want to delete, and choose **Delete snapshot**.







4. Choose **Yes** to confirm that you want to delete the snapshot.

Important

This is a permanent operation and can't be undone. You will lose all data on the snapshot when you delete it.

Delete multiple snapshots

1. From the Lightsail home page, choose **Snapshots**.
2. Find the Lightsail resource whose snapshots you want to delete, and choose the right-arrow to expand the list of snapshots.

 my-disk-for-windows-server-2012-r2 > 1 Disk Snapshot	8 GB Block Storage Disk – Oregon, all zones Last Snapshot: November 5, 2017 - 7:57 AM
 my-disk-for-wordpress-instance > 2 Disk Snapshot	64 GB Block Storage Disk – Oregon, all zones Last Snapshot: November 4, 2017 - 10:23 PM
 new-disk > 1 Disk Snapshot	64 GB Block Storage Disk – Oregon, all zones Last Snapshot: October 27, 2017 - 12:02 PM
 my-disk-for-windows-server > 1 Disk Snapshot	128 GB Block Storage Disk – Oregon, all zones Last Snapshot: November 5, 2017 - 7:57 AM

3. Choose **Delete multiple**.
4. Choose the snapshots you want to delete, and choose **Delete**.
5. Choose **Yes** to confirm that you want to delete the snapshots.

Important

This is a permanent operation and can't be undone. You will lose all data on the snapshots when you delete them.

Enable or disable automatic snapshots for Lightsail instances and disks

When you enable the automatic snapshots feature of your instance or block storage disk, Amazon Lightsail creates daily snapshots of your resource during the default automatic snapshot time, or during a [time you specify](#). Just like a manual snapshot, you can use an automatic snapshot as a baseline to create new resources or for data backup.

When automatic snapshots are created, you are billed the [snapshot storage fee](#) for the automatic snapshots stored on your Lightsail account.

Contents

- [Automatic snapshot restrictions](#)
- [Automatic snapshot retention](#)
- [Enable or disable automatic instance snapshots using the Lightsail console](#)
- [Enable or disable automatic snapshots for instances or block storage disks using the AWS CLI](#)

Automatic snapshot restrictions

The following restrictions apply to automatic snapshots:

- Automatic snapshots cannot be enabled or disabled for block storage disks using the Lightsail console. To enable or disable automatic snapshots for block storage disks, you must use the Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs. For more information, see [Enable or disable automatic snapshots using the AWS CLI](#).
- Automatic snapshot is currently not supported for Windows instances, or managed databases. Instead, you must create manual snapshots of your Windows instances or managed databases to back them up. For more information, see [Create a snapshot of your Windows Server instance](#) and [Create a database snapshot](#). Managed databases also have the point-in-time backup feature enabled by default, which you can use to restore your data to a new database. For more information, see [Create a database from a point-in-time backup](#).
- Automatic snapshots don't retain tags from the source resource. To keep a tag from the source resource on a new resource created from an automatic snapshot, you must manually add the tag when you create the new resource from the automatic snapshot. For more information, see [Add tags to a resource](#).

Automatic snapshot retention

The latest seven daily automatic snapshots are stored before the oldest one is replaced with the newest one. Additionally, all automatic snapshots associated with a resource are deleted when you delete the source resource. This behavior differs from manual snapshots, which are kept in your Lightsail account even after you delete the source resource. To keep automatic snapshots from being replaced, or from being deleted when you delete the source resource, you can [copy automatic snapshots as a manual snapshot](#).

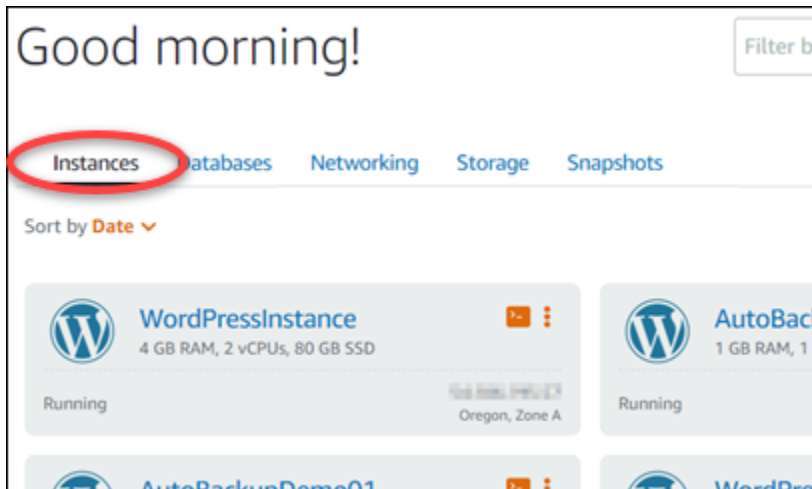
When you disable the automatic snapshot feature for a resource, the existing automatic snapshots of the resource are kept with the source resource until you do one of the following:

- Re-enable automatic snapshots and the existing automatic snapshots are replaced by newer snapshots.
- [Manually delete the existing automatic snapshots.](#)
- Delete the source resource, which deletes the associated automatic snapshots.

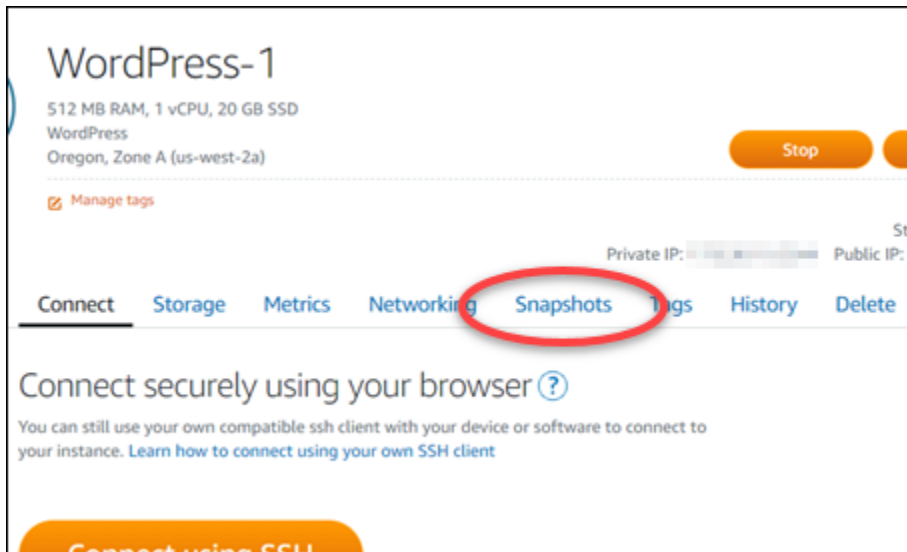
Enable or disable automatic instance snapshots using the Lightsail console

Complete the following steps to enable or disable automatic snapshots for an instance using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.



3. Choose the name of the instance for which you want to enable or disable automatic snapshots.
4. On the instance management page, choose the **Snapshots** tab.



5. Under the **Automatic snapshots** section, choose the toggle to enable it. Likewise, choose the toggle to disable it if it's enabled.
6. At the prompt, choose **Yes, enable** to enable automatic snapshots, or **Yes, disable** to disable the feature.

The automatic snapshot is enabled or disabled after a few moments.

- If you *enabled* the automatic snapshots feature, you may want to also change the automatic snapshot time. For more information, see [Change the automatic snapshot time for instances or block storage disks](#).
- If you *disabled* the automatic snapshots feature, the existing automatic snapshots of the resource are kept until you re-enable the feature and they are replaced by new snapshots, or until you delete them. You will be billed the [snapshot storage fee](#) for the automatic snapshots stored on your Lightsail account. For more information about deleting automatic snapshots, see [Delete automatic instance snapshots](#).

Enable or disable automatic snapshots for instances or block storage disks using the AWS CLI

Complete the following steps to enable or disable automatic snapshots for an instance or block storage disk using the AWS CLI.

1. Open a Terminal or Command Prompt window.

If you haven't already, [install the AWS CLI](#) and [configure it to work with Lightsail](#).

2. Enter one of the commands described in this step depending on whether you want to enable or disable automatic snapshots:

Note

The `autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00}` parameter is optional in these commands. If you don't specify a daily automatic snapshot time when you enable automatic snapshots, Lightsail assigns a default snapshot time for your resource. For more information, see [Change the automatic snapshot time for instances or block storage disks](#).

- Enter the following command to enable automatic snapshots for an existing resource:

```
aws lightsail enable-add-on --region Region --resource-name ResourceName --add-on-request  
addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00}
```

In the command, replace:

- *Region* with the AWS Region in which the resource is located.
- *ResourceName* with the name of the resource.
- *HH:00* with the daily automatic snapshot time in an hourly increment, and in Coordinated Universal Time (UTC).

Example:

```
aws lightsail enable-add-on --region us-west-2 --resource-name WordPress-1 --add-on-request  
addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=18:00}
```

- Enter the following command to enable automatic snapshots when creating a new instance:

```
aws lightsail create-instances --region Region --availability-zone AvailabilityZone --blueprint-id BlueprintID --  
bundle-id BundleID --instance-name InstanceName --add-ons  
addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00}
```

In the command, replace:

- *Region* with the AWS Region in which the instance should be created.
- *AvailabilityZone* with the availability zone in which the instance should be created.
- *BlueprintID* with the blueprint ID to use for the instance.
- *BundleID* with the bundle ID to use for the instance.
- *InstanceName* with the name to use for the instance.
- *HH:00* with the daily automatic snapshot time in an hourly increment, and in Coordinated Universal Time (UTC).

Example:

```
aws lightsail create-instances --region us-west-2 --availability-  
zone us-west-2a --blueprint-id wordpress_5_1_1_2 --bundle-  
id medium_2_0 --instance-name WordPressInstance --add-ons  
addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=20:00}
```

- Enter the following command to enable automatic snapshots when creating a new disk:

```
aws lightsail create-disk --region Region --availability-  
zone AvailabilityZone --size-in-gb Size --disk-name DiskName --add-ons  
addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00}
```

In the command, replace:

- *Region* with the AWS Region in which the disk should be created.
- *AvailabilityZone* with the availability zone in which the disk should be created.
- *Size* with the desired size of the disk in GB.
- *DiskName* with the name to use for the disk.
- *HH:00* with the daily automatic snapshot time in an hourly increment, and in Coordinated Universal Time (UTC).

Example:

```
aws lightsail create-disk --region us-west-2 --availability-  
zone us-west-2a --size-in-gb 32 --disk-name Disk01 --add-ons  
addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=18:59}
```

- Enter the following command to disable automatic snapshots for a resource:

```
aws lightsail disable-add-on --region Region --resource-name ResourceName --add-on-type AutoSnapshot
```

In the command, replace:

- *Region* with the AWS Region in which the resource is located.
- *ResourceName* with the name of the resource.

Example:

```
aws lightsail disable-add-on --region us-west-1 --resource-name MyFirstWordPressWebsite01 --add-on-type AutoSnapshot
```

You should see a result similar to the following example:

```
{
  "operations": [
    {
      "id": "2610213c-d68f-488e-9124-245913a2a22a",
      "resourceName": "WordPressInstance",
      "resourceType": "Instance",
      "createdAt": 1566431564.323,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "CreateInstance",
      "status": "Started",
      "statusChangedAt": 1566431564.323
    },
    {
      "id": "fd04446d-8106-4c7e-8d69-f42be811453a",
      "resourceName": "WordPressInstance",
      "resourceType": "Instance",
      "createdAt": 1566431566.368,
      "location": {
        "availabilityZone": "us-west-2",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "EnableAddOn - AutoBackup",
      "operationType": "EnableAddOn",
      "status": "Started"
    }
  ]
}
```

The automatic snapshot is enabled or disabled after a few moments.

- If you *enabled* automatic snapshots, you may want to also change the automatic snapshot time. For more information, see [Change the automatic snapshot time for instances or block storage disks](#).

- If you *disabled* automatic snapshots, the existing automatic snapshots are kept until you re-enable the feature and they are replaced by new snapshots, or until you delete them. You will be billed the [snapshot storage fee](#) for the automatic snapshots stored on your Lightsail account. For more information about deleting automatic snapshots, see [Delete automatic instance snapshots](#).

Note

For more information about the EnableAddOn and DisableAddOn API operations in these commands, see [EnableAddOn](#) and [DisableAddOn](#) in the Lightsail API documentation.

Change the time for automatic snapshots in Lightsail

When you [enable the automatic snapshots feature](#) for an instance or block storage disk, Lightsail creates daily snapshots of the resource during the [default automatic snapshot time](#), or a time you specify. Follow the steps in this guide to change the automatic snapshot time for your resource.

Contents

- [Automatic snapshot time restrictions](#)
- [Default automatic snapshot times for AWS Regions](#)
- [Change the automatic snapshot time using the Lightsail console](#)
- [Change the automatic snapshot time and block storage disks using the AWS CLI](#)

Automatic snapshot time restrictions

The following restrictions apply to the automatic snapshot time:

- The automatic snapshot time cannot be changed for block storage disks using the Lightsail console. To change the automatic snapshot time for block storage disks, you must use the Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs. For more information, see [Change the automatic snapshot time using the AWS CLI](#).

- The automatic snapshot time can be specified only in hourly increments. It also must be a time that is more than 30 minutes from your current time. Lightsail creates the automatic snapshot between the time you specify and up to 45 minutes after.

⚠ Important

You cannot create manual snapshots when an automatic snapshot is being created.

- When you change the automatic snapshot time for a resource, it is typically effective immediately, except under the following conditions:
 - If an automatic snapshot has been created for the current day, and you change the snapshot time to a later time of day, then the new snapshot time will be effective the following day. This ensures that two snapshots are not created for the current day.
 - If an automatic snapshot has not yet been created for the current day, and you change the snapshot time to an earlier time of day, then the new snapshot time will be effective the following day. Also, a snapshot is automatically created at the previously set time for the current day. This ensures that a snapshot is created for the current day.
 - If an automatic snapshot has not yet been created for the current day, and you change the snapshot time to a time that is within 30 minutes from your current time, then the new snapshot time will be effective the following day. Also, a snapshot is automatically created at the previously set time for the current day. This ensures that a snapshot is created for the current day, because 30 minutes is required between your current time and the new snapshot time that you specify.
 - If an automatic snapshot is scheduled to be created within 30 minutes from your current time and you change the snapshot time, then the new snapshot time will be effective the following day. Also, a snapshot is automatically created at the previously set time for the current day. This ensures that a snapshot is created for the current day, because 30 minutes is required between your current time and the new snapshot time that you specify.

When any of these conditions are true, a message displays in the Lightsail console to notify you that the new snapshot time may take up to 24 hours to take effect.

Default automatic snapshot times for AWS Regions

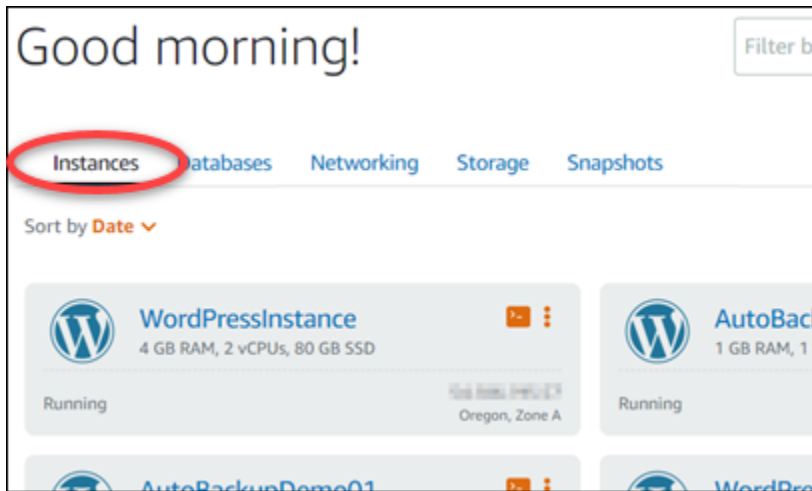
If you don't specify an automatic snapshot time when you enable automatic snapshots, then Lightsail assigns one of the following default automatic snapshot times. The times depend on the AWS Region where your instance or block storage disk is located:

- US East (Ohio) (us-east-2): 03:00 UTC
- US East (N. Virginia) (us-east-1): 06:00 UTC
- US West (Oregon) (us-west-2): 06:00 UTC
- Asia Pacific (Mumbai) (ap-south-1): 17:00 UTC
- Asia Pacific (Seoul) (ap-northeast-2): 13:00 UTC
- Asia Pacific (Singapore) (ap-southeast-1): 14:00 UTC
- Asia Pacific (Sydney) (ap-southeast-2): 12:00 UTC
- Asia Pacific (Tokyo) (ap-northeast-1): 13:00 UTC
- Canada (Central) (ca-central-1): 06:00 UTC
- EU (Frankfurt) (eu-central-1): 20:00 UTC
- EU (Ireland) (eu-west-1): 22:00 UTC
- EU (London) (eu-west-2): 06:00 UTC
- EU (Paris) (eu-west-3): 07:00 UTC
- EU (Stockholm) (eu-north-1): 08:00 UTC

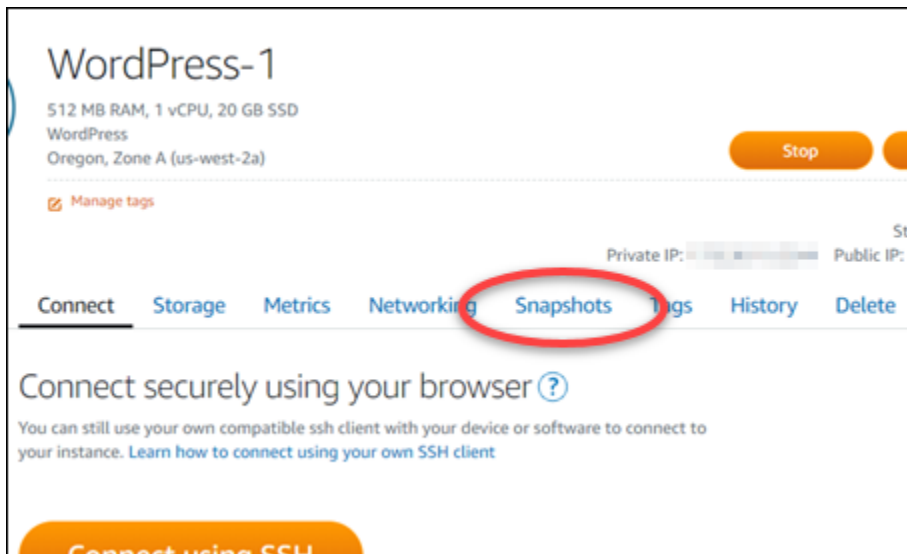
Change the automatic snapshot time using the Lightsail console

Complete the following steps to change the automatic snapshot time for an instance using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.



3. Choose the name of the instance for which you want to change the automatic snapshot time.
4. On the instance management page, choose the **Snapshots** tab.



5. Under the **Automatic snapshots** section, choose **Change snapshot time**.
6. Choose a time of day when you'd like Lightsail to create an automatic snapshot. The time that you choose must be in Coordinated Universal Time (UTC).
7. Choose **Change** to save the new snapshot time.

The automatic snapshot time is updated after a few moments. A restriction may apply to the effective date of your new automatic snapshot time. For more information, see [Automatic snapshot time restrictions](#).

Change the automatic snapshot time for instances and block storage disks using the AWS CLI

Complete the following steps to change the automatic snapshot time for an instance or block storage disk using the AWS CLI.

1. Open a Terminal or Command Prompt window.

If you haven't already, [install the AWS CLI](#) and [configure it to work with Lightsail](#).

2. Enter the following command to change the automatic snapshot time for a resource:

```
aws lightsail enable-add-on --region Region --resource-name ResourceName --add-on-request addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00}
```

In the command, replace:

- *Region* with the AWS Region in which the resource is located.
- *ResourceName* with the name of the resource.
- *HH:00* with the daily automatic snapshot time in an hourly increment, and in Coordinated Universal Time (UTC).

Example:

```
aws lightsail enable-add-on --region us-west-1 --resource-name MyFirstWordPressWebsite01 --add-on-request addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=12:00}
```

You should see a result similar to the following example:

```
{
  "operation": {
    "id": "enable-add-on-auto-backup-1",
    "resourceName": "WordPress-1",
    "resourceType": "Instance",
    "createdAt": 1566501867.165,
    "location": {
      "availabilityZone": "us-west-2",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "EnableAddOn - AutoBackup",
    "operationType": "EnableAddOn",
    "status": "Started"
  }
}
```

The automatic snapshot time is updated after a few moments. A restriction may apply to the effective date of your new automatic snapshot time. For more information, see [Automatic snapshot time restrictions](#).

Note

For more information about the EnableAddOn API operation in this command, see [EnableAddOn](#) in the Lightsail API documentation.

Delete automatic snapshots in Lightsail

You can delete automatic snapshots of an instance or block storage disk in Amazon Lightsail at any time; whether the feature is enabled, or if it's disabled after it had been enabled. You will be billed the [snapshot storage fee](#) for the automatic snapshots stored on your Lightsail account. Follow the steps in this guide to delete automatic snapshots if you no longer need them. For example, if you've [copied an automatic snapshot to a manual snapshot](#) and you no longer need the original, or if you've [disabled the automatic snapshots feature](#) for your resource and you don't need the existing automatic snapshots that were kept.

Contents

- [Delete automatic snapshots restriction](#)
- [Delete automatic snapshots of an instance using the Lightsail console](#)
- [Delete automatic snapshots of an instance or block storage disk using the AWS CLI](#)

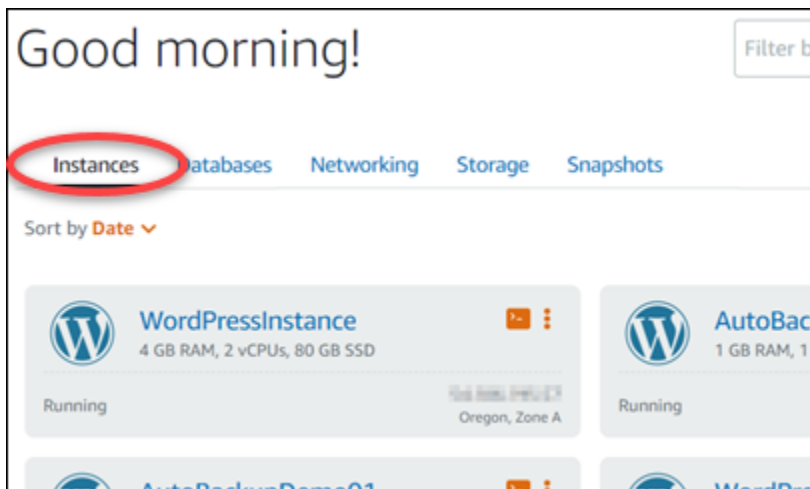
Delete automatic snapshots restriction

Automatic snapshots of block storage disks cannot be deleted using the Lightsail console. To delete an automatic snapshot of a block storage disk, you must use the Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs. For more information, see [Delete automatic snapshots of an instance or block storage disk using the AWS CLI](#).

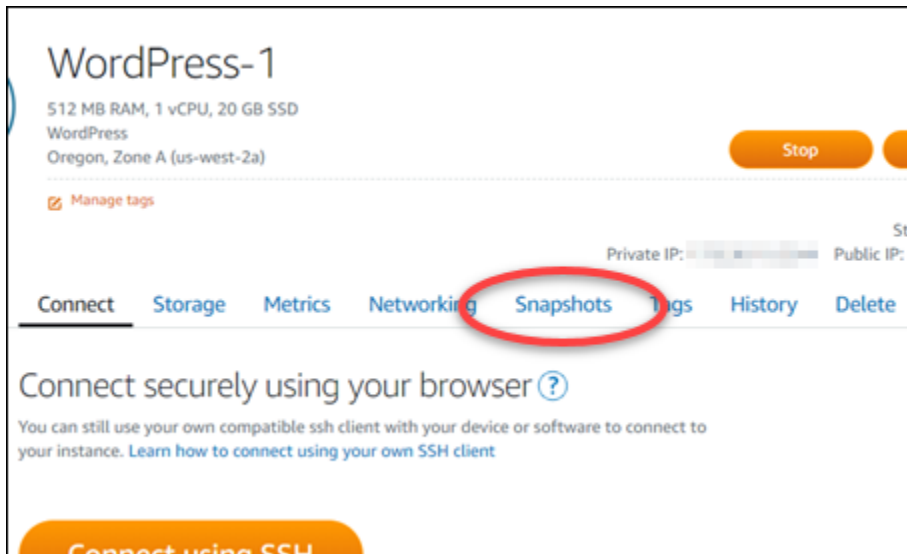
Delete automatic snapshots of an instance using the Lightsail console

Complete the following steps to delete automatic snapshots of an instance using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.



3. Choose the name of the instance for which you want to delete automatic snapshots.
4. On the instance management page, choose the **Snapshots** tab.



5. Under the **Automatic snapshots** section, choose the ellipsis icon next to the automatic snapshot that you want to delete, then choose **Delete snapshot**.
6. At the prompt, choose **Yes** to confirm that you want to delete the snapshot.

The automatic snapshot is deleted after a few moments.

Delete automatic snapshots of an instance or block storage disk using the AWS CLI

Complete the following steps to delete automatic snapshots of an instance or block storage disk using the AWS CLI.

1. Open a Terminal or Command Prompt window.

If you haven't already, [install the AWS CLI](#) and [configure it to work with Lightsail](#).

2. Enter the following command to get the dates of the available automatic snapshots for a specific resource. You will need the date of the automatic snapshot to specify as the date parameter in the subsequent command.

```
aws lightsail --region Region get-auto-snapshots --resource-name ResourceName
```

In the command, replace:

- *Region* with the AWS Region in which the resource is located.
- *ResourceName* with the name of the resource.

Example:

```
aws lightsail --region us-west-2 get-auto-snapshots --resource-name MyFirstWordPressWebsite01
```

You should see a result similar to the following, which lists the available automatic snapshots:

```
{
  "resourceName": "Magento-2",
  "resourceType": "Instance",
  "autoBackups": [
    {
      "date": "2019-08-22",
      "createdAt": 1566455335.0,
      "status": "Success",
      "fromAttachedDisks": [
        {
          "path": "/dev/xvdf",
          "sizeInGb": 8
        }
      ]
    },
    {
      "date": "2019-08-21",
      "createdAt": 1566368935.0,
      "status": "Success",
      "fromAttachedDisks": []
    },
    {
      "date": "2019-08-20",
      "createdAt": 1566282535.0,
      "status": "Success",
      "fromAttachedDisks": []
    },
    {
      "date": "2019-08-19",
      "createdAt": 1566196135.0,
      "status": "Success",
      "fromAttachedDisks": []
    }
  ]
}
```

3. Enter the following command to delete an automatic snapshot:

```
aws lightsail --region Region delete-auto-snapshot --resource-name ResourceName --date YYYY-MM-DD
```

In the command, replace:

- *Region* with the AWS Region in which the resource is located.

- *ResourceName* with the name of the resource.
- *YYYY-MM-DD* with the date of the available auto snapshot that you obtained using the preceding command.

Example:

```
aws lightsail --region us-west-2 delete-auto-snapshot --resource-  
name MyFirstWordPressWebsite01 --date 2019-09-16
```

You should see a result similar to the following example:

```
{  
  "operation": {  
    "id": "8f253c00-c34f-4073-9b0e-e5507ce264d9",  
    "resourceName": "Magento-2",  
    "resourceType": "Instance",  
    "createdAt": 1566507472.323,  
    "location": {  
      "availabilityZone": "us-west-2",  
      "regionName": "us-west-2"  
    },  
    "isTerminal": true,  
    "operationDetails": "DeleteAutoBackup-2019-08-16",  
    "operationType": "DeleteAutoBackup",  
    "status": "Succeeded"  
  }  
}
```

The automatic snapshot is deleted after a few moments.

Note

For more information about the `GetAutoSnapshots` and `DeleteAutoSnapshot` API operations in these commands, see [GetAutoSnapshots](#) and [DeleteAutoSnapshot](#) in the Lightsail API documentation.

Keep automatic snapshots in Lightsail

When you [enable the automatic snapshots feature](#) for an instance or block storage disk in Amazon Lightsail, only the latest seven daily automatic snapshots of the resource are stored. Then, the

oldest one is replaced with the newest one. Additionally, all automatic snapshots associated with a resource are deleted when you delete the source resource.

If you want to keep a specific automatic snapshot from being replaced, or from being deleted when you delete the source resource, you can copy it as a manual snapshot. Manual snapshots are kept until you manually delete them.

Follow the steps in this guide to keep an automatic snapshot by copying it as a manual snapshot. You will be billed the [snapshot storage fee](#) for the automatic snapshots stored on your Lightsail account.

Note

If you disable the automatic snapshots feature for a resource, the existing automatic snapshots of the resource are kept until you re-enable the feature and they are replaced by newer snapshots, or you until you [delete the automatic snapshots](#).

Contents

- [Keep automatic snapshots restriction](#)
- [Keep automatic snapshots of instances using the Lightsail console](#)
- [Keep automatic snapshots of instances and block storage disks using the AWS CLI](#)

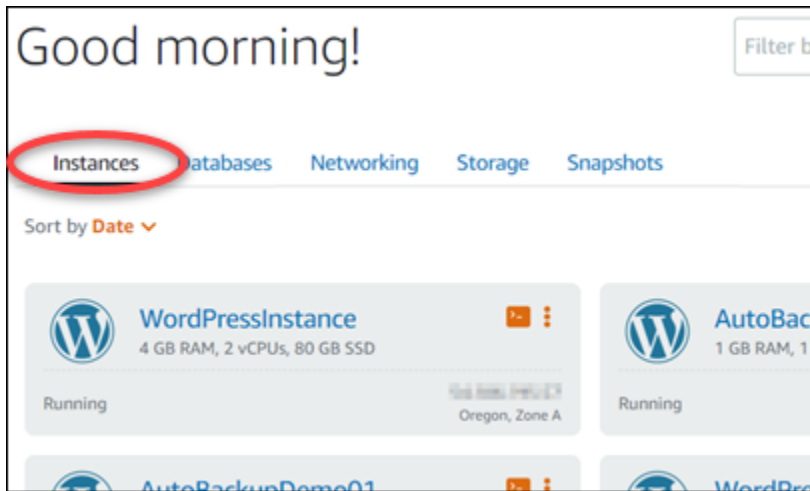
Keep automatic snapshots restriction

Automatic snapshots of block storage disks cannot be copied to manual snapshots using the Lightsail console. To copy an automatic snapshot of a block storage disk, you must use the Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs. For more information, see [Keep automatic snapshots of instances and block storage disks using the AWS CLI](#).

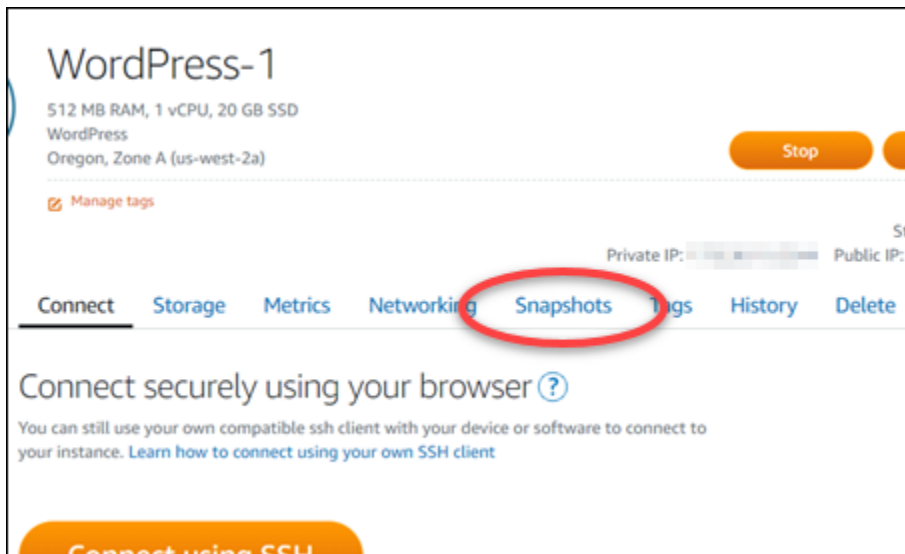
Keep automatic snapshots of instances using the Lightsail console

Complete the following steps to keep automatic snapshots for an instance using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.



3. Choose the name of the instance for which you want to keep automatic snapshots.
4. On the instance management page, choose the **Snapshots** tab.



5. Under the **Automatic snapshots** section, choose the ellipsis icon next to the automatic snapshot that you want to keep, then choose **Keep snapshot**.
6. At the prompt, choose **Yes, save** to confirm that you want to keep the automatic snapshot.

The automatic snapshot is copied as a manual snapshot after a few moments. Manual snapshots are kept until you delete them.

⚠ Important

If you no longer need the automatic snapshot, we recommend that you delete it. Otherwise, you will be billed the [snapshot storage fee](#) for the automatic snapshot and

the duplicate manual snapshot stored on your Lightsail account. For more information, see [Delete automatic instance snapshots](#).

Keep automatic snapshots of instances and block storage disks using the AWS CLI

Complete the following steps to keep automatic snapshots for an instance or block storage disk using the AWS CLI.

1. Open a Terminal or Command Prompt window.

If you haven't already, [install the AWS CLI](#) and [configure it to work with Lightsail](#).

2. Enter the following command to get the dates of the available automatic snapshots for a specific resource. You need the date of the automatic snapshot to specify as the `restore date` parameter in the subsequent command.

```
aws lightsail get-auto-snapshots --region Region --resource-name ResourceName
```

In the command, replace:

- *Region* with the AWS Region in which the resource is located.
- *ResourceName* with the name of the resource.

Example:

```
aws lightsail get-auto-snapshots --region us-west-2 --resource-name MyFirstWordPressWebsite01
```

You should see a result similar to the following, which lists the available automatic snapshots:

```
{
  "resourceName": "Magento-2",
  "resourceType": "Instance",
  "autoBackups": [
    {
      "date": "2019-08-22",
      "createdAt": 1566455335.0,
      "status": "Success",
      "fromAttachedDisks": [
        {
          "path": "/dev/xvdf",
          "sizeInGb": 8
        }
      ]
    },
    {
      "date": "2019-08-21",
      "createdAt": 1566368935.0,
      "status": "Success",
      "fromAttachedDisks": []
    },
    {
      "date": "2019-08-20",
      "createdAt": 1566282535.0,
      "status": "Success",
      "fromAttachedDisks": []
    },
    {
      "date": "2019-08-19",
      "createdAt": 1566196135.0,
      "status": "Success",
      "fromAttachedDisks": []
    }
  ]
}
```

3. Enter the following command to keep an automatic snapshot for a specific resource:

```
aws lightsail copy-snapshot --region TargetRegion --source-resource-
name ResourceName --restore-date YYYY-MM-DD --source-region SourceRegion --target-
snapshot-name SnapshotName
```

In the command, replace:

- *TargetRegion* with the AWS Region in which you want to copy the snapshot to.
- *ResourceName* with the name of the resource.
- *YYYY-MM-DD* with the date of the available auto snapshot that you obtained using the preceding command.
- *SourceRegion* with the AWS Region in which the automatic snapshot is currently in.
- *SnapshotName* with the name of the new snapshot to be created.

Example:

```
aws lightsail copy-snapshot --region us-west-2 --source-resource-  
name MyFirstWordPressWebsite01 --restore-date 2019-09-16 --source-region us-west-2  
--target-snapshot-name Snapshot-Copied-From-Auto-Snapshot
```

You should see a result similar to the following example:

```
{  
  "operations": [  
    {  
      "id": "6f2607ca-c3d3-4e92-9795-8d7c8d72b038",  
      "resourceName": "Snapshot-Copied-From-Auto-Backup",  
      "resourceType": "InstanceSnapshot",  
      "createdAt": 1566504306.107,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationDetails": "us-west-2:Magento-2",  
      "operationType": "CopySnapshot",  
      "status": "Started",  
      "statusChangedAt": 1566504306.107  
    }  
  ]  
}
```

The automatic snapshot is copied as a manual snapshot after a few moments. Manual snapshots are kept until you delete them.

 Important

If you no longer need the automatic snapshot, we recommend that you delete it. Otherwise, you will be billed the [snapshot storage fee](#) for the automatic snapshot and duplicate manual snapshot stored on your Lightsail account. For more information, see [Delete automatic instance snapshots](#).

Note

For more information about the `GetAutoSnapshots` and `CopySnapshot` API operations in these commands, see [GetAutoSnapshots](#) and [CopySnapshot](#) in the Lightsail API documentation.

Copy Lightsail snapshots from one AWS Region to another

Amazon Lightsail lets you copy instance snapshots and block storage disk snapshots from one AWS Region to another, or within the same Region. Copy snapshots between Regions if you created and configured resources in one Region, but later decide that a different Region is more appropriate. Or, if you want to replicate your resources across multiple Regions. This guide describes the process of copying Lightsail snapshots.

Prerequisites

Create a snapshot of the Lightsail instance or block storage disk that you want to copy. For more information, see one of the following guides:

- [Create a snapshot of your Linux or Unix instance](#)
- [Create a snapshot of your Windows Server instance](#)
- [Create a block storage disk snapshot](#)

Copy a snapshot

You can copy Lightsail instance snapshots and block storage disk snapshots from one AWS Region to another, or within the same Region.

To copy a Lightsail snapshot

1. Sign in to the [Lightsail console](#).
2. From the Lightsail home page, choose the **Snapshots** tab.
3. Locate the instance or block storage disk that you want to copy, and expand the node to view the available snapshots for that resource.

- Choose the actions menu icon (⋮) for the desired snapshot, then choose **Copy to another Region**.

The screenshot shows the AWS Management Console 'Snapshots' page. The navigation tabs include 'Instances', 'Databases', 'Networking', 'Storage', and 'Snapshots'. The page is sorted by 'Region' and then by 'Date'. Under the 'Virginia (us-east-1)' region, there are two instance snapshots:

- Amazon_Linux-512MB-Virginia-1**: 512 MB RAM, 1 vCPU, 20 GB SSD – Virginia, all zones (us-east-1). It has 1 instance snapshot. The last snapshot is from November 30, 2018, at 2:26 PM. A context menu is open over this snapshot, showing options: 'Create new instance', 'Copy to another Region' (highlighted), 'Export to Amazon EC2', and 'Delete snapshot'.
- Windows_Server_2016-512MB-Virgini...**: 512 MB RAM, 1 vCPU, 30 GB SSD – Virginia, all zones (us-east-1). It has 2 instance snapshots. The last snapshot is from November 30, 2018, at 2:26 PM.

- On the **Copy a snapshot** page, in the **Snapshot to copy** section, confirm that the snapshot details displayed match the specifications of the source instance or block storage disk.

The screenshot shows the 'Snapshot to copy' page. It states 'You are making a copy of the following snapshot:'. The selected snapshot is:

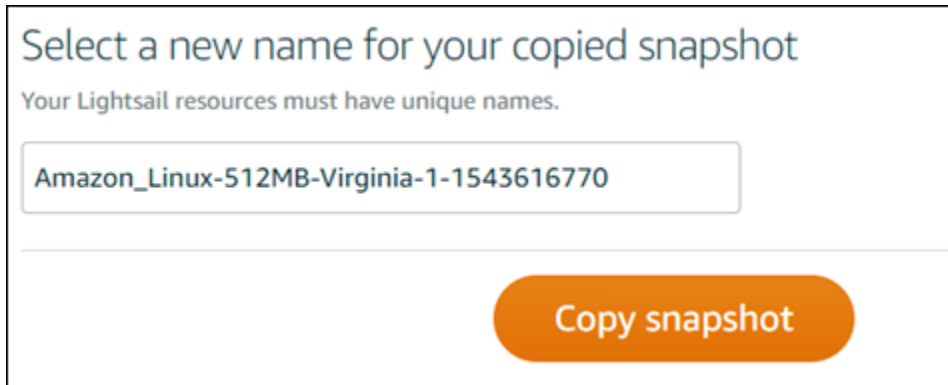
- Amazon_Linux-512MB-Virginia-1-1543616770**: November 30, 2018 - 2:26 PM. 512 MB RAM, 1 vCPU, 20 GB SSD, instance snapshot.

- In the **Select a Region** section of the page, choose the Region for your snapshot copy.
- Enter a name for your snapshot copy.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

8. Choose **Copy snapshot**.



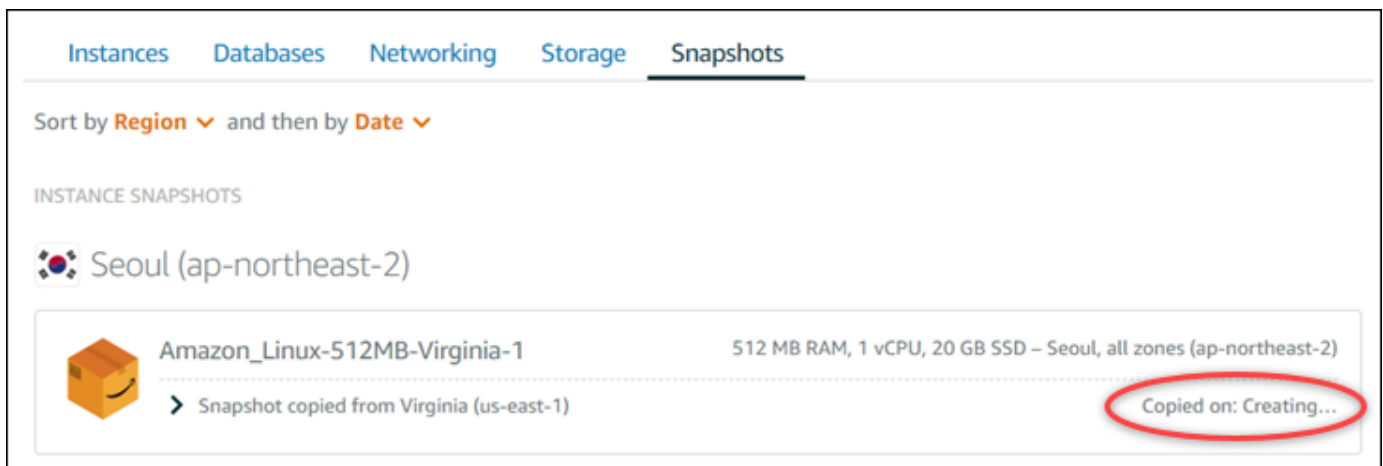
Select a new name for your copied snapshot

Your Lightsail resources must have unique names.

Amazon_Linux-512MB-Virginia-1-1543616770

Copy snapshot

Your snapshot copy should be available soon. It depends on the size and configuration of the source instance. You can check the status of your snapshot copy by browsing to the **Snapshots** tab on the Lightsail home page, and looking for the snapshot with a status of **Creating** as shown in the following screenshot. The status will change when the snapshot is ready.




Instances Databases Networking Storage **Snapshots**

Sort by **Region** ▾ and then by **Date** ▾

INSTANCE SNAPSHOTS

🇰🇷 Seoul (ap-northeast-2)

	Amazon_Linux-512MB-Virginia-1	512 MB RAM, 1 vCPU, 20 GB SSD – Seoul, all zones (ap-northeast-2)
	> Snapshot copied from Virginia (us-east-1)	Copied on: Creating...

Next steps

Here are a few additional steps you can perform after copying a snapshot to another Region in Lightsail:

- Create a new instance from the copied snapshot after it's available. For more information, see [Create an instance from a snapshot](#).
- Delete the source snapshot if you no longer need it. Otherwise, you will be billed for storing the snapshot.

Export Lightsail snapshots to Amazon EC2

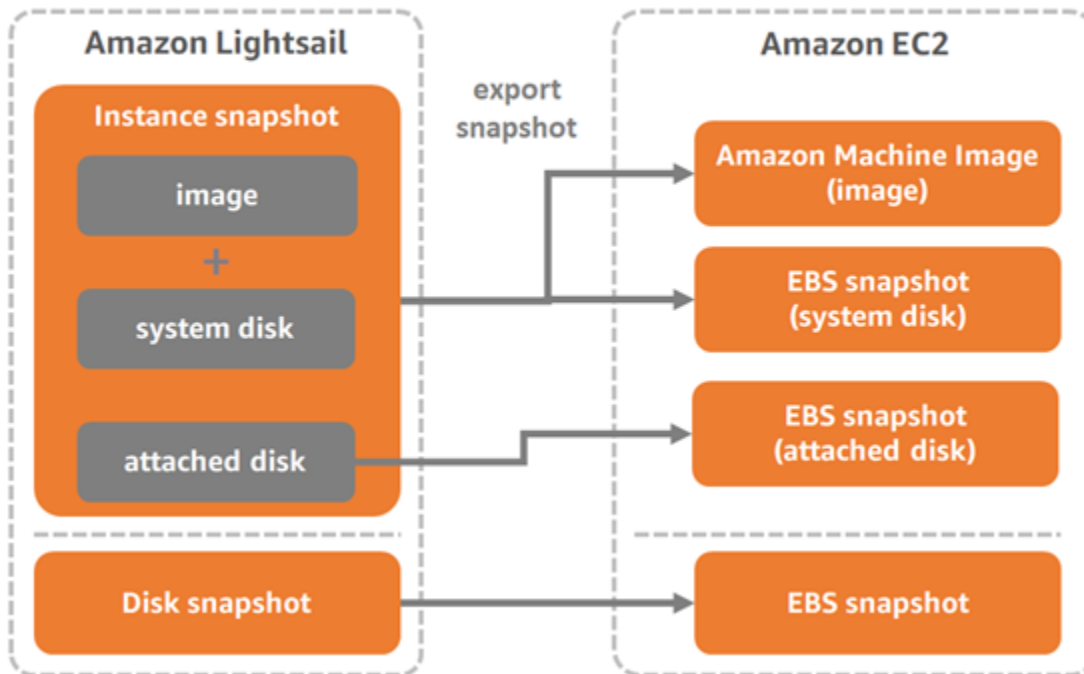
Lightsail instance and block storage disk snapshots can be exported to Amazon EC2 using one of the following methods:

- The Lightsail console. For more information, see [Export snapshots to Amazon EC2](#).
- The Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs. For more information, see the [ExportSnapshot operation](#) in the Lightsail API documentation, or the [export-snapshot command](#) in the AWS CLI documentation.

You can export instance snapshots and block storage disk snapshots. However, snapshots of Django, Ghost, and cPanel & WHM instances cannot be exported at this time. Snapshots are exported to the same AWS Region from Lightsail to Amazon EC2. To export snapshots to a different Region, first copy the snapshot to a different Region in Lightsail, then perform the export. For more information, see [Copy snapshots from one AWS Region to another](#).

Exporting a Lightsail instance snapshot results in an Amazon Machine Image (AMI) and an Amazon Elastic Block Store (Amazon EBS) snapshot being created in Amazon EC2. This is because Lightsail instances are comprised of an image and a system disk, but both are grouped together as a single instance entity in the Lightsail console to make them more efficient to manage. If the source Lightsail instance had one or more block storage disks attached to it when the snapshot was created, then additional EBS snapshots for each attached disk will be created in Amazon EC2. Exporting a Lightsail block storage disk snapshot results in a single EBS snapshot being created in Amazon EC2. All exported resources in Amazon EC2 have their own distinct unique identifiers that are different than their Lightsail counterparts.

Export Lightsail snapshots to Amazon EC2



Note

Lightsail uses an AWS Identity and Access Management (IAM) service-linked role (SLR) to export snapshots to Amazon EC2. For more information about SLRs, see [Service-linked roles](#).

The export process can take a while. It depends on the size and configuration of the source instance or block storage disk. Use the task monitor in the Lightsail console to track the status of your export. For more information, see [Task monitor](#).

Create Amazon EC2 resources from exported Lightsail snapshots

After a Lightsail snapshot is exported and available in Amazon EC2 (as an AMI, EBS snapshot, or both), you can create Amazon EC2 resources from the snapshot using one of the following methods:

- The **Create an Amazon EC2 instance** page in the Lightsail console, also known as the Upgrade to Amazon EC2 Wizard. For more information, see [Create Amazon EC2 instances from exported snapshots](#).

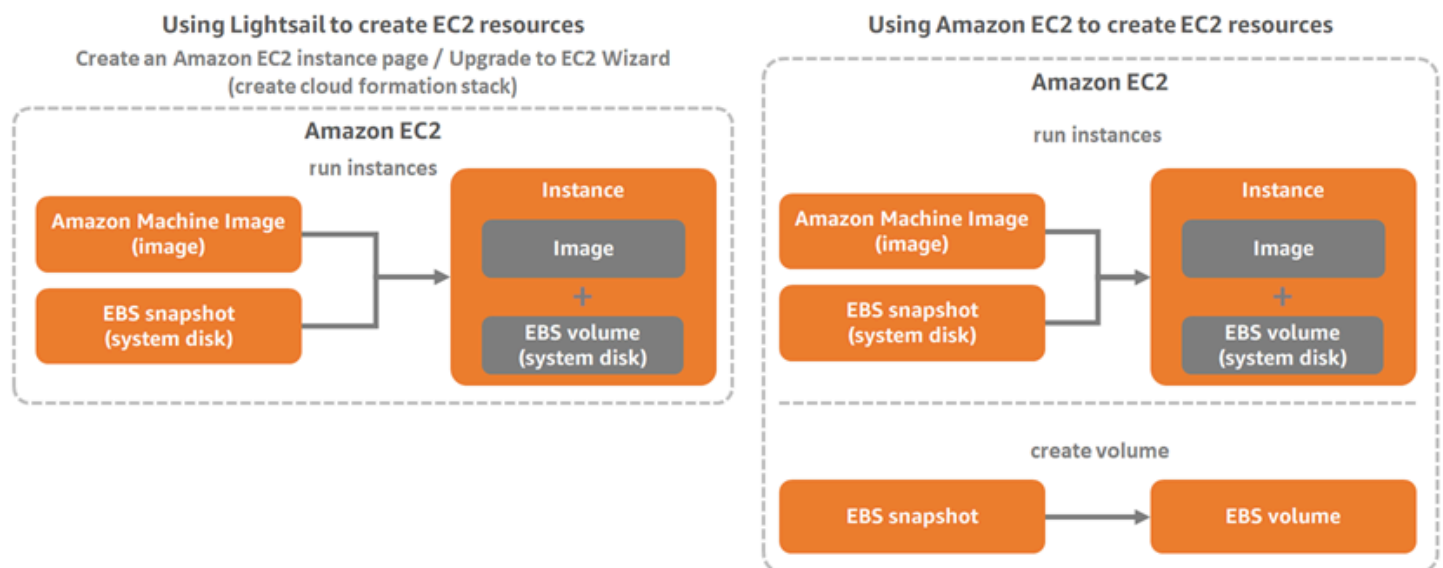
- The Lightsail API, AWS CLI, or SDKs. For more information, see the [CreateCloudFormationStack operation](#) in the Lightsail API documentation, or the [create-cloud-formation-stack command](#) in the AWS CLI documentation.

Note

Lightsail can be used to create Amazon EC2 instances from exported instance snapshots, but it cannot be used to create EBS volumes from exported block storage disk snapshots. For this, you must use the Amazon EC2 console, API, or AWS CLI. For more information, see [Create Amazon EBS volumes from exported disk snapshots](#).

- The Amazon EC2 console, Amazon EC2 API, AWS CLI, or SDKs. For more information, see [Launching an Instance Using the Launch Instance Wizard](#) or [Restoring an Amazon EBS Volume from a Snapshot](#) in the Amazon EC2 documentation.

Creating an Amazon EC2 instance from an exported instance snapshot (AMI and EBS snapshot) results in a single EC2 instance being launched. The AMI and EBS snapshot that resulted from exporting the Lightsail instance snapshot are automatically linked together to form the EC2 instance. The exported Lightsail block storage disk snapshot (EBS snapshot) can be used to create an EBS volume in Amazon EC2.



Note

Lightsail uses a CloudFormation stack to create instances and their related resources in EC2. For more information, see [AWS CloudFormation stacks for Lightsail](#).

The process to create Amazon EC2 resources from an exported snapshot can take a while. It depends on the size and configuration of the source instance. Use the task monitor in the Lightsail console to track the status of this task. For more information, see [Task monitor](#).

Choosing an Amazon EC2 instance type

Amazon EC2 offers a wider range of instance options than are available in Lightsail. In Amazon EC2, you can choose instance types that are optimized for compute (C5), memory (R5), or a balance of both (T3 and M5). Lightsail provides these options in the **Create an Amazon EC2 instance** page; however, more instance type options are available if you use Amazon EC2 to create new instances from an exported snapshot. For more information about EC2 instance types, see [Instance Types](#) in the Amazon EC2 documentation.

Before you create EC2 instances from exported snapshots, it is important to understand the instance price differences between Lightsail and Amazon EC2. For more information about instance pricing, see the [Lightsail pricing](#) and [Amazon EC2 pricing](#) pages.

Lightsail and Amazon EC2 instance type compatibility

Some Lightsail instances are incompatible with the current generation EC2 instance types (T3, M5, C5, or R5) because they are not enabled for enhanced networking. If your source Lightsail instance is incompatible, you will need to choose a previous generation instance type (T2, M4, C4, or R4) when creating an EC2 instance from your exported snapshot. These options are presented to you when creating an EC2 instance using the **Create an Amazon EC2 instance** page in the Lightsail console.

To use the latest generation EC2 instance types when the source Lightsail instance is incompatible, you need to create the new EC2 instance using a previous generation instance type (T2, M4, C4, or R4), update the networking driver, and then upgrade the instance to the desired current generation instance type. For more information, see [Enhanced networking for Amazon EC2 instances](#).

Connect to Amazon EC2 instances

You can connect to Amazon EC2 instances similar to how you connect to Lightsail instances. This means using SSH for Linux and Unix instances and RDP for Windows Server instances. However, the browser-based SSH/RDP client that you might have used in the Lightsail console might not be available in Amazon EC2 depending on the browser version that you're using, so you may need to configure your own SSH/RDP client to connect to your EC2 instances. For more information, see the following guides:

- [Connect to an Amazon EC2 Linux or Unix instance that was created from a Lightsail snapshot](#)
- [Connect to an Amazon EC2 Windows Server instance that was created from a Lightsail snapshot](#)

Secure an Amazon EC2 instance

After you create an EC2 instance from an exported Lightsail snapshot, you may need to perform a few actions to improve the security of your new instances. The actions are different depending on the operating system of your EC2 instance.

Securing Linux and Unix instances in Amazon EC2

If you create a Linux or Unix instance in Amazon EC2 from an exported snapshot using EC2 (the EC2 console, the EC2 API, AWS CLI for EC2, or SDKs for EC2), the new EC2 instance may contain residual SSH keys from the Lightsail service. We recommend removing these keys to better secure the new instance.

For more information, see [Secure an Amazon EC2 Linux or Unix instance that was created from a Lightsail snapshot](#).

Securing Windows Server instances in Amazon EC2

After you create a Windows Server instance in Amazon EC2 from an exported snapshot, any user in your AWS account with access to Lightsail and EC2 will be able to retrieve the default administrator password first assigned to the source instance, which is also the password for the new EC2 instance. For increased security, we recommend that you change the default administrator password for your Amazon EC2 instance, if you haven't already done so.

For more information, see [Secure an Amazon EC2 Windows Server instance that was created from a Lightsail snapshot](#).

Export Lightsail snapshots and create resources in Amazon EC2

To get started with exporting snapshots, and creating Amazon EC2 resources from them, see the following guides:

- [Task monitor](#)
- [AWS CloudFormation stacks for Lightsail](#)
- [Export snapshots to Amazon EC2](#)
- [Create Amazon EC2 instances from exported snapshots](#)
- [Create Amazon EBS volumes from exported disk snapshots](#)
- [Enhanced networking for Amazon EC2 instances](#)
- [Connect to an Amazon EC2 Linux or Unix instance that was created from a Lightsail snapshot](#)
- [Connect to an Amazon EC2 Windows Server instance that was created from a Lightsail snapshot](#)
- [Secure an Amazon EC2 Linux or Unix instance that was created from a Lightsail snapshot](#)
- [Secure an Amazon EC2 Windows Server instance that was created from a Lightsail snapshot](#)
- [Copy snapshots from one AWS Region to another](#)
- [Service-linked roles](#)

How to export Lightsail snapshots to Amazon EC2

You can export Amazon Lightsail instance and block storage disk snapshots to Amazon Elastic Compute Cloud (Amazon EC2). Exporting a Lightsail instance snapshot results in an Amazon Machine Image (AMI) and an Amazon Elastic Block Store (Amazon EBS) snapshot being created in Amazon EC2. This is because Lightsail instances are comprised of an image and a system disk, but both are grouped together as a single instance entity in the Lightsail console to make them more efficient to manage. If the source Lightsail instance has one or more block storage disks attached to it when the snapshot is created, then additional EBS snapshots for each attached disk are created in Amazon EC2.

Exporting a Lightsail block storage disk snapshot results in a single EBS snapshot being created in Amazon EC2. All exported resources in Amazon EC2 have their own distinct unique identifiers that are different than their Lightsail counterparts.

This guide describes how to export a Lightsail snapshot, track the status of your export, and the next steps after the exported snapshot is available in Amazon EC2 (as an AMI, EBS snapshot, or both).

Important

We recommend getting familiar with the Lightsail export process before completing the steps in this guide. For more information, see [Export snapshots to Amazon EC2](#).

Contents

- [Service-linked role and required IAM permissions to export Lightsail snapshots](#)
- [Prerequisites](#)
- [Export a Lightsail snapshot to Amazon EC2](#)
- [Track the status of your export](#)

Service-linked role and required IAM permissions to export Lightsail snapshots

Lightsail uses an AWS Identity and Access Management (IAM) service-linked role (SLR) to export snapshots to Amazon EC2. For more information about SLRs, see [Service-linked roles](#).

The following additional permissions may need to be configured in IAM depending on the user that will perform the snapshot export:

- If the [Amazon account root user](#) will perform the export, then continue to the [Prerequisites section](#) of this guide. The account root user already has the required permissions to perform the snapshot export.
- If an IAM user will perform the export, then an AWS account administrator must add the following policy to the user. For more information about how to change permissions for a user, see [Changing Permissions for an IAM User](#) in the IAM documentation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
```

```
        "Resource": "arn:aws:iam::*:role/aws-service-role/
lightsail.amazonaws.com/AWSServiceRoleForLightsail*",
        "Condition": {"StringLike": {"iam:AWSServiceName":
"lightsail.amazonaws.com"}}
    },
    {
        "Effect": "Allow",
        "Action": "iam:PutRolePolicy",
        "Resource": "arn:aws:iam::*:role/aws-service-role/
lightsail.amazonaws.com/AWSServiceRoleForLightsail*"
    }
]
}
```

Prerequisites

Create a snapshot of the Lightsail instance or block storage disk that you want to export to Amazon EC2. For more information, see one of the following guides:

- [Create a snapshot of your Linux or Unix instance](#)
- [Create a snapshot of your Windows Server instance](#)
- [Create a block storage disk snapshot](#)

Export a Lightsail snapshot to Amazon EC2

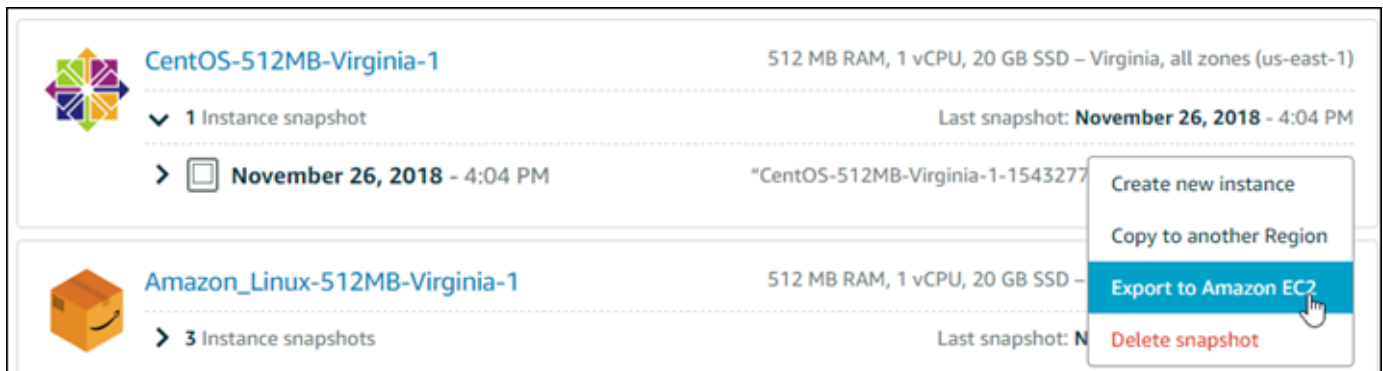
The most efficient way to export a snapshot to Amazon EC2 is by using the Lightsail console. You can also export snapshots using the Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs. For more information, see the [ExportSnapshot operation](#) in the Lightsail API documentation, or the [export-snapshot command](#) in the AWS CLI documentation.

Note

Snapshots are exported to the same AWS Region from Lightsail to Amazon EC2. To export snapshots to a different Region, first copy the snapshot to a different Region in Lightsail, then perform the export. For more information, see [Copy snapshots from one AWS Region to another](#).

To export a Lightsail snapshot to Amazon EC2

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Snapshots** tab.
3. Locate the instance or block storage disk that you want to export, and expand the node to view the available snapshots for that resource.
4. Choose the **Action** menu for the desired snapshot, then choose **Export to Amazon EC2**.



Note

Snapshots of cPanel & WHM, Django, and Ghost instances cannot be exported to Amazon EC2 at this time.

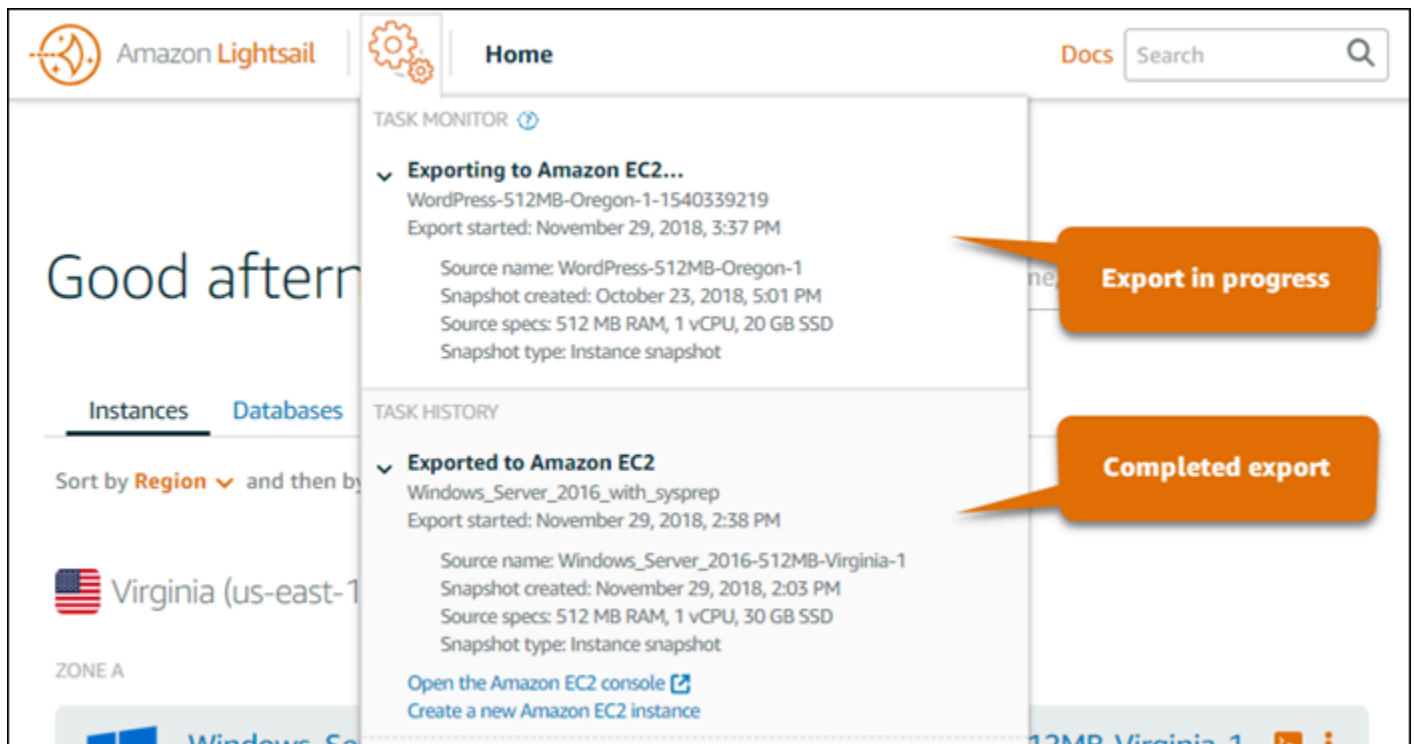
5. Review the important details displayed on the prompt.
6. If you agree to export to Amazon EC2, choose **Yes, continue** to begin the process.

The export process can take a while. It depends on the size and configuration of the source instance or block storage disk. Continue to the [Track the status of your export](#) section of this guide to track the status of your export.

Track the status of your export

Use the task monitor in the Lightsail console to track the status of your export. It can be accessed from the top navigation pane on all pages of the Lightsail console. For more information, see [Task monitor](#).

The following information is displayed in the task monitor for snapshot exports:



- **Snapshot name** — The name of the source Lightsail snapshot.
- **Export started** — The date and time the snapshot export was started.
- **Snapshot created** — The date and time the source Lightsail snapshot was created.
- **Source specs** — The specifications of the source Lightsail instance, such as the memory, processing, and storage.
- **Snapshot type** — The type of the Lightsail snapshot. It's either an instance snapshot or disk snapshot.

The following information is displayed in the task monitor for completed snapshot exports:

- **Exported** is displayed if the snapshot was successfully exported to Amazon EC2.
- **Failed** is displayed if there was a problem exporting the snapshot.

If the snapshot was successfully exported, then the task monitor displays the following options for the completed export:

- **Create a new Amazon EC2 instance** — Choose this option to create a new instance in Amazon EC2 using the Lightsail console. For more information, see [Create Amazon EC2 instances from exported snapshots](#).

- **Open the Amazon EC2 console** — Choose this option to use the Amazon EC2 console to create new EC2 resources from your exported snapshot. If you exported a Lightsail block storage disk snapshot, then you must use Amazon EC2 to create an EBS volume from the snapshot (an EBS snapshot). For more information, see [Launching an Instance Using the Launch Instance Wizard](#) or [Restoring an Amazon EBS Volume from a Snapshot](#) in the Amazon EC2 documentation.

Note

Delete the source Lightsail snapshot if you no longer need it. Otherwise, you will be billed for storing it.

Create Amazon EBS volumes from exported Lightsail disk snapshots

After a Lightsail block storage disk snapshot is exported and available in Amazon EC2 (as an EBS snapshot), you can create an EBS volume from the snapshot using the Amazon EC2 console.

Note

To create EC2 instances from exported instance snapshots, see [Creating Amazon EC2 instances from exported snapshots in Lightsail](#).

You can also create new EBS volumes using the Amazon EC2 API, AWS CLI, or SDKs. For more information, see [Launch an Instance Using the Launch Instance Wizard](#) or [Restoring an Amazon EBS Volume from a Snapshot](#) in the Amazon EC2 documentation.

Important

We recommend getting familiar with the Lightsail export process before completing the steps in this guide. For more information, see [Export snapshots to Amazon EC2](#).

Prerequisites

Export a Lightsail block storage disk snapshot to Amazon EC2. For more information, see [Export snapshots to Amazon EC2](#).

Create an EBS volume from an exported Lightsail block storage disk snapshot

Use the Amazon EC2 console to create a new EBS volume from an exported Lightsail block storage disk snapshot.

Note

These steps are also in the Amazon EC2 documentation. To learn more, see [Restoring an Amazon EBS Volume from a Snapshot](#) in the Amazon EC2 documentation.

To create an EBS volume from an exported Lightsail block storage disk snapshot

1. Sign in to the [Amazon EC2 console](#).
2. From the navigation bar, select the region that your snapshot is located in.
3. In the navigation pane, choose **Elastic Block Store**, and then choose **Snapshots**.
4. Locate and select the exported Lightsail block storage disk snapshot.

Exported disk snapshot can be identified by the *A disk snapshot exported from Amazon Lightsail* description of the EBS snapshot as shown in the following screenshot:

Snapshot ID	Size	Description
snap-0c8daaae6d815c3f7	20 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...
snap-06bbbf02cdbe92137	30 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...
snap-044c549df2bf34f5e	8 GiB	A disk snapshot exported from Amazon Lightsail MyDiskSnapshot
snap-01fe78a3c611911ed	20 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...
snap-0c635b87c5675cb8d	8 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...
snap-0964d597917e3487d	30 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...
snap-054c5c705820b90e1	8 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...
snap-0a80ad5fd849fcd1b	20 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...
snap-0042eb3868771694d	20 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...
snap-014a072c2a77360bb	8 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...
snap-0c0f05832bd08a09b	8 GiB	A disk snapshot exported from Amazon Lightsail MyDiskSnapshot
snap-0763258cc2b12f96a	20 GiB	Copied for DestinationPool and-0Lc78829f6d3111f60 from SourcePool and-0e1b...

5. Choose **Actions**, then choose **Create Volume**.

6. Choose a volume type from the **Volume Type** drop-down menu. For more information, see [Amazon EBS Volume Types](#) in the Amazon EC2 documentation.
7. For **Size (GiB)**, type the size of the volume, or verify that the default size of the snapshot is adequate.
8. With a Provisioned IOPS SSD volume, for **IOPS**, type the maximum number of input/output operations per second (IOPS) that the volume should support.
9. For **Availability Zone**, choose the Availability Zone in which to create the volume. EBS volumes can only be attached to EC2 instances in the same Availability Zone.
10. (Optional) Choose **Create additional tags** to add tags to the volume. For each tag, provide a tag key and a tag value.
11. Choose **Create Volume**. After your volume is created, it is listed in the **Elastic Block Store > Volumes** section of the Amazon EC2 console.

Next steps

Here are a few additional steps you can perform after creating a new Amazon EC2 instance:

- After you've restored a volume from a snapshot, you can attach it to an instance to begin using it. For more information, see [Attaching an Amazon EBS Volume to an Instance](#) in the Amazon EC2 documentation.
- If you restored a snapshot to a larger volume than the default for that snapshot, you must extend the file system on the volume to take advantage of the extra space. For more information, see [Modifying the Size, IOPS, or Type of an EBS Volume on Linux](#) in the Amazon EC2 documentation.

Create Amazon EC2 instances from exported Lightsail snapshots

After a Lightsail instance snapshot is exported and available in Amazon EC2 (as an AMI and an EBS snapshot), you can create an Amazon EC2 instance from the snapshot using the **Create an Amazon EC2 instance** page in the Amazon Lightsail console, also known as the Upgrade to Amazon EC2 wizard. It guides you through the EC2 instance configuration options, such as choosing an EC2 instance type that matches your requirements, configuring your security group ports, adding a launch script, and more. The wizard in the Lightsail console simplifies the process of creating new EC2 instances and their related resources.

Note

To create Amazon Elastic Block Store (Amazon EBS) volumes from exported block storage disk snapshots, see [Create Amazon EBS volumes from exported disk snapshots](#).

You can also create new EC2 instances using the Lightsail API, AWS CLI, or SDKs. For more information, see the [CreateCloudFormationStack operation](#) in the Lightsail API documentation, or the [create-cloud-formation-stack command](#) in the AWS CLI documentation. Or if you're comfortable with Amazon EC2, you can use the EC2 console, Amazon EC2 API, AWS CLI, or SDKs. For more information, see [Launching an Instance Using the Launch Instance Wizard](#) or [Restoring an Amazon EBS Volume from a Snapshot](#) in the Amazon EC2 documentation.

Important

We recommend getting familiar with the Lightsail export process before completing the steps in this guide. For more information, see [Export snapshots to Amazon EC2](#).

Contents


- [AWS CloudFormation stack for Lightsail](#)
- [Prerequisites](#)
- [Access the Create an Amazon EC2 instance page in the Lightsail console](#)
- [Create an Amazon EC2 instance](#)
- [Track the status of your new Amazon EC2 instance](#)
- [Next steps](#)

AWS CloudFormation stack for Lightsail

Lightsail uses an AWS CloudFormation stack to create EC2 instances and their related resources. For more information about the CloudFormation stacks for Lightsail, see [AWS CloudFormation stacks for Lightsail](#).

The following additional permissions may need to be configured in IAM depending on the user that will create the EC2 instance using the **Create an Amazon EC2 instance** page:

- If the [Amazon account root user](#) will create the EC2 instance, then continue to the [Prerequisites section](#) of this guide. The root user already has the required permissions to create EC2 instances using Lightsail.
- If an IAM user will create the EC2 instance, then an AWS account administrator must add the following permissions to the user. For more information about how to change permissions for a user, see [Changing Permissions for an IAM User](#) in the IAM documentation.
- The following permissions are required for users to create Amazon EC2 instances using Lightsail:

 **Note**

These permissions allow the CloudFormation stack to be created. However, if the creation fails, the rollback process might require more permissions. Lack of permissions may lead to remaining resources not rolled back in Amazon EC2. If this happens, you can go to the AWS CloudFormation console and manually delete the EC2 resources. For more information, see [AWS CloudFormation stacks for Lightsail](#)

- ec2:DescribeAvailabilityZones
- ec2:DescribeSubnets
- ec2:DescribeRouteTables
- ec2:DescribeInternetGateways
- ec2:DescribeVpcs
- cloudformation:CreateStack
- cloudformation:ValidateTemplate
- iam:CreateServiceLinkedRole
- iam:PutRolePolicy
- The following permissions are required if the user will configure ports in the security group for the EC2 instance:
 - ec2:DescribeSecurityGroups
 - ec2:CreateSecurityGroup
 - ec2:AuthorizeSecurityGroupIngress
- The following permissions are required if the user is creating a Windows Server instance in

- `ec2:DescribeKeyPairs`
- `ec2:ImportKeyPair`
- The following permissions are required if the user is creating Amazon EC2 instances for the first time, or when the virtual private cloud (VPC) fails to configure completely:
 - `ec2:AssociateRouteTable`
 - `ec2:AttachInternetGateway`
 - `ec2:CreateInternetGateway`
 - `ec2:CreateRoute`
 - `ec2:CreateRouteTable`
 - `ec2:CreateSubnet`
 - `ec2:CreateVpc`
 - `ec2:ModifySubnetAttribute`
 - `ec2:ModifyVpcAttribute`

Prerequisites

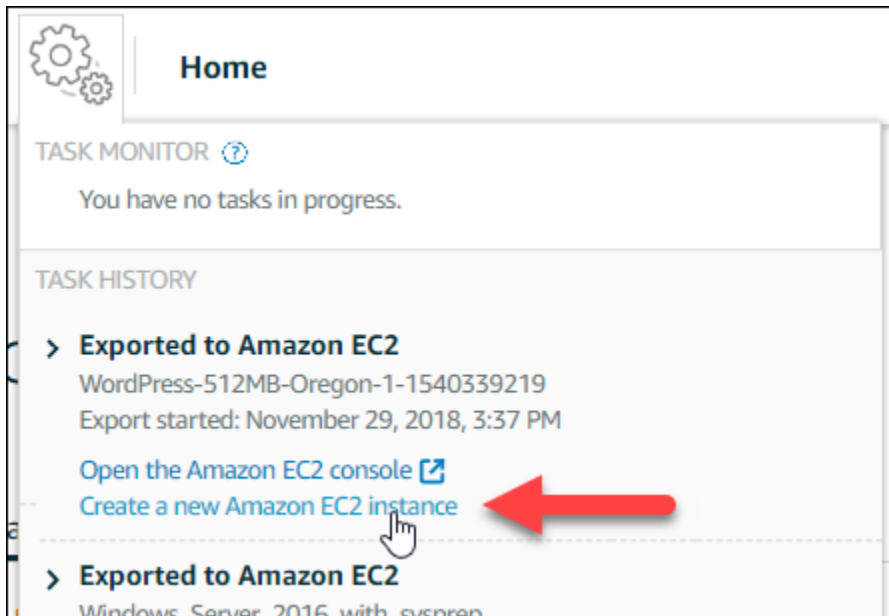
Export a Lightsail instance snapshot to Amazon EC2. For more information, see [Export snapshots to Amazon EC2](#).

Access the Create an Amazon EC2 instance page in the Lightsail console

The **Create an Amazon EC2 instance** page in the Lightsail console can be accessed from the task monitor only after an instance snapshot is successfully exported to EC2.

To access the Create an Amazon EC2 instance page in the Lightsail console

1. Sign in to the [Lightsail console](#).
2. From the top navigation pane, choose the **Task monitor** icon.
3. Locate the completed instance snapshot export in the **Task history** section, then choose **Create a new Amazon EC2 instance**.



The **Create an Amazon EC2 instance** page appears. Continue to the following [Create an Amazon EC2 instance](#) section of this guide to learn how to configure and create an EC2 instance using this page.

Create an Amazon EC2 instance

Use the **Create an Amazon EC2 instance** page to create an EC2 instance. To create more than one EC2 instance from an exported Lightsail snapshot, repeat the following steps multiple times but wait until each instance is created before creating the next one.

To create an Amazon EC2 instance

1. On the **Amazon EC2 AMI details** section of the page, confirm that the Amazon Machine Image (AMI) details displayed match the specifications of the source Lightsail instance.

Amazon EC2 AMI details




WordPress-512MB-Oregon-1

"WordPress-512MB-Oregon-1-1540339219 "

512 MB RAM, 1 vCPU, 20 GB SSD, Amazon EC2 AMI

Including 1 attached disk:

 20 GB SSD System Disk

2. On the **Resource location** section of the page, change the Availability Zone of your instance if necessary. The Amazon EC2 resources are created in the same AWS Region as the source Lightsail snapshot.

Note

Not all Availability Zones may be available for all users. Choosing an unavailable Availability Zone will result in an error when creating the EC2 instance.

Resource location



You are creating this EC2 instance in **Oregon, Zone A** (us-west-2a)

 [Change zone](#)



Amazon EC2 uses a different zone letter mapping than Lightsail.

Your preferred zone for Oregon (us-west-2) may not be available.

3. On the **Compute resource** section of the page, choose one of the following options:

- a. **Find closest match** to automatically select an Amazon EC2 instance type that closely matches the specifications of the source Lightsail instance.
- b. **Help me choose** to answer a quick questionnaire about the specifications of your new Amazon EC2 instance. You can select from instance types that are compute optimized, memory optimized, or balanced between the two.
- c. **Select manually** to view a list of instance types available through the **Create an Amazon EC2 instance** page.

Note


Some Lightsail instances are incompatible with the current generation EC2 instance types (T3, M5, C5, or R5) because they are not enabled for enhanced networking. If your source Lightsail instance is incompatible, you will need to choose a previous generation instance type (T2, M4, C4, or R4) when creating an EC2 instance from your exported snapshot. These instance type options are presented to you on the **Create an Amazon EC2 instance** page in the Lightsail console.

To use the latest generation EC2 instance types when the source Lightsail instance is incompatible, you need to create the new EC2 instance using a previous generation instance type (T2, M4, C4, or R4), update the networking driver, and then upgrade the instance to the desired current generation instance type. For more information, see [Update Amazon EC2 instances for enhanced networking](#).


4. On the **Optional** section of the page:

OPTIONAL


The firewall port configuration for your Amazon EC2 instance are configured in the instance's security group.

 **Specify port configuration**

You can add a shell script that will run on your instance the first time it launches.

 **Add launch script**

- a. Choose **Specify port configuration** to select the firewall settings for your Amazon EC2 instance, then choose one of the following options:

Security groups 

How would you like to configure the security group for your Amazon EC2 instance?

Use the default firewall settings from the Lightsail image.

Use the source Lightsail instance firewall settings.

The following open ports will be imported into the security group for your EC2 instance:

APPLICATION	PROTOCOL	PORT RANGE
SSH	TCP	22
HTTP	TCP	80
HTTPS	TCP	443

- i. **Use the default firewall settings from the Lightsail image** to configure the default ports from the source Lightsail blueprint on your new EC2 instance. For more information about the default ports for Lightsail blueprints, see [Firewalls and ports](#).
- ii. **Use the source Lightsail instance firewall settings** to configures the ports from the source Lightsail instance on your new EC2 instance. This option is only available when the source Lightsail instance is still active.
- b. On the **Launch script** section of the page, choose **Add launch script** if you wish to add a script that configures your EC2 instance when it launches.
5. On the **Connection security** section of the page, determine how you connected to the source Lightsail instance. This ensures that you get the correct SSH key to connect to your new EC2 instance. You may have connected to the source Lightsail instance using one of the following methods:

- a. **Using the default Lightsail key pair for the source instance's region** — Download and use the unique default Lightsail key for that AWS Region to connect to your EC2 instance.

Note

The default Lightsail key pair is always used on Windows Server instances in Lightsail.

- b. **Using your own key pair** — Locate the private key and use it to connect to your EC2 instance.


Note

Lightsail does not store your personal private keys. Therefore; the option to download your private key is not provided. If you are unable to locate your private key, then you will not be able to connect to your EC2 instance.


6. On the **Storage resources** section of the page, confirm that the EBS volumes being created match the system disk and any attached block storage disks for the source Lightsail instance.

Storage resources ?

We will create **2** EBS volumes for you and link them to your instance



Storage volume
/dev/xvdf
8 GB General Purpose (GP2) Encrypted EBS Volume



System volume
/dev/xvda
20 GB General Purpose (GP2) Encrypted EBS Volume

7. Review the important details about creating resources outside of Lightsail.
8. If you agree to create the instance in Amazon EC2, choose **Create resources in EC2**.

Lightsail confirms that your instance is being created, and information about the AWS CloudFormation stack is displayed. Lightsail uses a CloudFormation stack to create the EC2 instance and its related resources. For more information, see [AWS CloudFormation stacks for Lightsail](#).

Continue to the [Track the status of your new Amazon EC2 instance](#) section of this guide to track the status of your new EC2 instance.

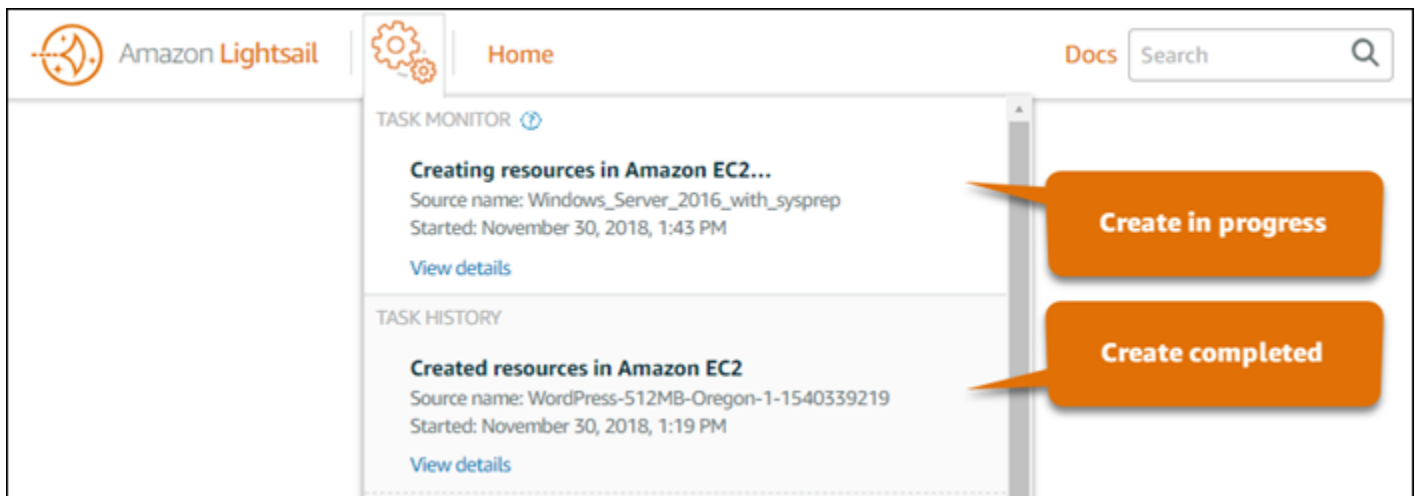
Important

Wait until after your new EC2 instance is created to create another EC2 instance from the same exported snapshot.

Track the status of your new Amazon EC2 instance

Use the task monitor in the Lightsail console to track the status of your new EC2 instance. It can be accessed from the top navigation pane on all pages of the Lightsail console. For more information, see [Task monitor](#).

The following information is displayed in the task monitor for EC2 instances being created:



The screenshot shows the Amazon Lightsail console interface. At the top, there is a navigation bar with the Amazon Lightsail logo, a 'Home' button, and a search bar. Below the navigation bar, the 'TASK MONITOR' section is visible. It contains two task entries:

- Creating resources in Amazon EC2...**
Source name: Windows_Server_2016_with_sysprep
Started: November 30, 2018, 1:43 PM
[View details](#)
- Created resources in Amazon EC2**
Source name: WordPress-512MB-Oregon-1-1540339219
Started: November 30, 2018, 1:19 PM
[View details](#)

Two orange callout boxes are overlaid on the right side of the task monitor. The first callout points to the first task and contains the text 'Create in progress'. The second callout points to the second task and contains the text 'Create completed'.

- **Source name** — The name of the source Lightsail snapshot.
- **Started** — The date and time that the create request was started.

The following information is displayed in the task monitor for EC2 instances that have been created:

- **Created** is displayed if the Amazon EC2 resources were successfully created. Continue to the [Next steps](#) section of this guide for the next steps after your new EC2 instance is ready.
- **Failed** is displayed if there was a problem creating EC2 instance.

Next steps

Here are a few additional steps you can perform after creating an Amazon EC2 instance:

- You can connect to Amazon EC2 instances similar to how you connect to Lightsail instances. This means using SSH for Linux and Unix instances and RDP for Windows Server instances. However, the browser-based SSH/RDP client that you might have used in the Lightsail console might not be available in Amazon EC2 depending on the browser version that you're using, so you might need to configure your own SSH/RDP client to connect to your EC2 instances. For more information, see the following guides:
 - [Connect to an Amazon EC2 Linux or Unix instance that was created from a Lightsail snapshot](#)
 - [Connect to an Amazon EC2 Windows Server instance that was created from a Lightsail snapshot](#)
- Linux or Unix instances in Amazon EC2 created from Lightsail snapshots may contain residual SSH keys from Lightsail. We recommend removing these keys to better secure your EC2 instance. For more information, see [Secure an Amazon EC2 Linux or Unix instance that was created from a Lightsail snapshot](#).

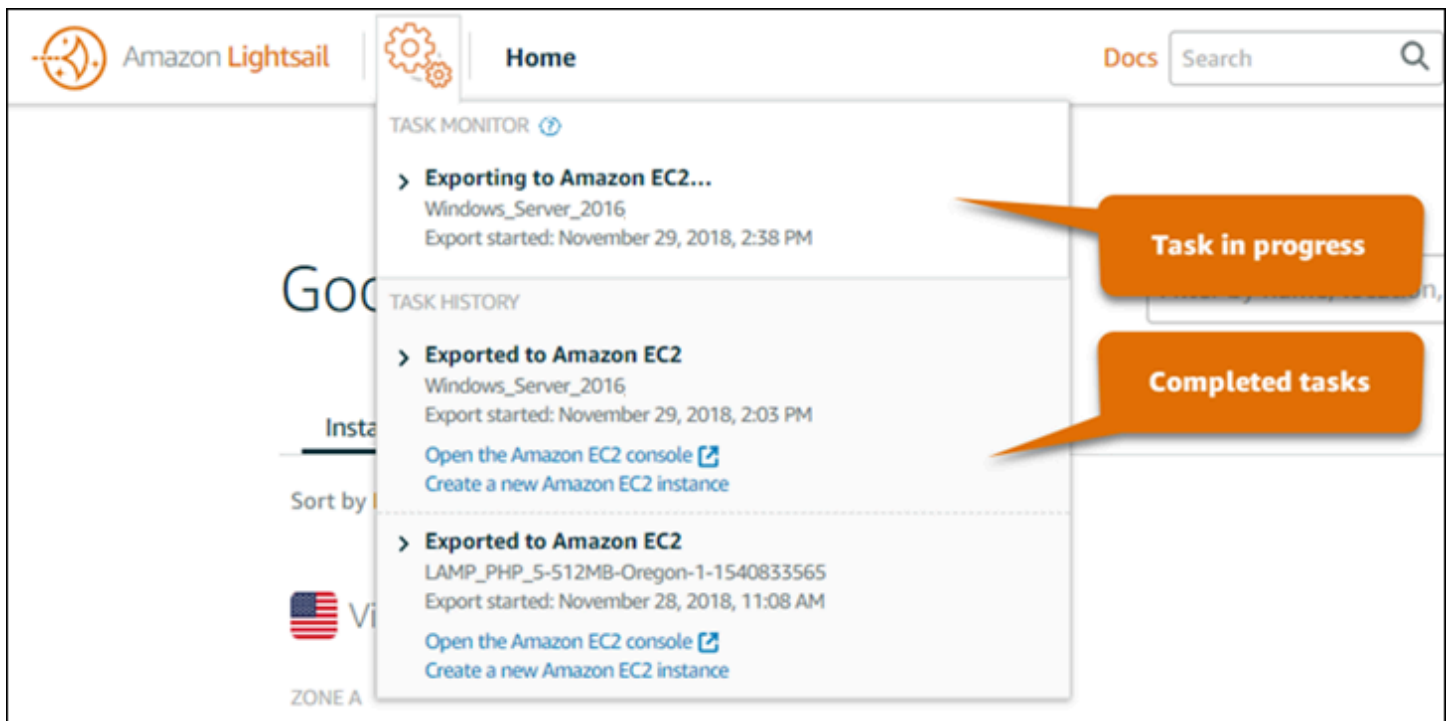
After your EC2 instance is created, you might need to perform a few more steps so that it's configured the same way as the source Lightsail instance. Here are a few additional steps to configure your EC2 instance:

- Configure the firewall settings by editing the security group for your Amazon EC2 instance. For more information, see [Amazon EC2 Security Groups for Linux Instances](#) or [Amazon EC2 Security Groups for Windows Instances](#) in the Amazon EC2 documentation.
- If you created a Lightsail static IP, and attached it to your Lightsail instance, then you should create and attach an Elastic IP to your Amazon EC2 instance. For more information, see [Elastic IP Addresses](#) in the Amazon EC2 documentation.

- If you created a Lightsail DNS zone, and configured a domain for your Lightsail instance, then you should create an Amazon Route 53 DNS zone, use it to manage your domain's DNS, and point your domain to your new Amazon EC2 instance. For more information, see [Configuring Amazon Route 53 as Your DNS Service and Making Amazon Route 53 the DNS Service for an Existing Domain](#) in the Amazon Route 53 documentation.
- If you created a Lightsail load balancer, and configured it for your Lightsail instances, then you should configure an Application Load Balancer for your Amazon EC2 instances. For more information, see [Getting Started with Application Load Balancers](#) in the Elastic Load Balancing documentation.
- Lightsail databases cannot be accessed by Amazon EC2 instances. If the Lightsail instance you exported to Amazon EC2 is connected to a Lightsail database; then you will need to manually migrate that database to Amazon Relational Database Service (Amazon RDS) to access its data from the new Amazon EC2 instance. For more information, see [Importing Data to an Amazon RDS MySQL or MariaDB DB Instance with Reduced Downtime](#) and [Connecting to an Amazon RDS DB Instance](#).

Lightsail console task monitor

The task monitor in the Amazon Lightsail console tracks the status of exporting Lightsail snapshots to Amazon EC2, or creating new EC2 instances from exported instance snapshots. These tasks can take a while depending on the size and configuration of the source instance or block storage disk. The task monitor displays the latest 20 tasks that are in progress or completed. It can be accessed from the top navigation pane on all pages of the Lightsail console. The task monitor icon is orange when a task is in progress, or grey when all tasks are completed.



For more information about exporting Lightsail snapshots to Amazon EC2, or creating EC2 instances from exported snapshots, see the following guides:

- [Export snapshots to Amazon EC2](#)
- [Create Amazon EC2 instances from exported snapshots](#)

Domain registration in Amazon Lightsail

Your website needs a name, such as `example.com`. Amazon Lightsail lets you register a name for your website, known as a domain name. To access your website, users type your domain name into their web browser.

Use the **Domains & DNS** tab in the Amazon Lightsail console to register and manage domain names. Lightsail uses Amazon Route 53, a highly available and scalable Domain Name System (DNS) web service, to register domains for you. After your domain is registered, you can assign it to your Lightsail resources or manage DNS records for it. For general information about DNS, see [DNS](#).

For more information about domain registration in Amazon Lightsail, continue reading.

Contents

- [How domain registration works](#)
- [Domains that you can register in Lightsail](#)
- [Pricing for domain registration](#)

How domain registration works

The following overview shows how you register a domain name in Amazon Lightsail:

1. Confirm that the domain name you want is available to use on the internet. If the domain name you want is not available, you can try other names or change only the top-level domain, such as **.com**, to another top-level domain, such as **.org** or **.net**. For a list of the top-level domains (TLDs) that Lightsail supports, see [Domains that you can register in Amazon Lightsail](#).
2. Register the domain name with Lightsail. When you register a domain, you provide names and contact information for the domain owner and other contacts.

At the end of the registration process, we send the information that you provide to the registrar for the domain. The domain registrar is a company that is accredited by the Internet Corporation for Assigned Names and Numbers (ICANN) to process domain registrations for specific TLDs. The registrar for the domain is either Amazon Registrar or our registrar associate, Gandi.

Amazon Registrar and Gandi hide different information by default. Amazon Registrar, Inc. hides all of your contact information, and Gandi hides all of your contact information except organization name.

- To find out who the registrar is for your domain, see [Domains that you can register in Amazon Lightsail](#).
- The registrar sends your information to the registry for the domain. A registry is a company that sells domain registrations for one or more top-level domains, such as **.com**.
- The registry stores the information about your domain in their own database and also stores some of the information in the public WHOIS database.

For more information about how to register a domain name, see [Register a new domain](#).

After you register a domain using Lightsail, Route 53 makes itself the DNS service for your domain by assigning a set of name servers to your domain. A name server is a server that helps translate domain names into IP addresses. .

Lightsail automatically does the following to make itself the DNS service for the domain:

- Creates a [Lightsail DNS zone](#) that has the same name as your domain.
- Assigns a set of four name servers to the Lightsail DNS zone.
- Replaces the domain's Route 53 name servers with the name servers from your Lightsail DNS zone.

If you already registered a domain name with another registrar, you can choose to transfer management of the domain's DNS to Lightsail. This isn't required to use other Lightsail features. For more information, see [Create a DNS zone to manage your domain's DNS records](#).

Domains that you can register in Lightsail

Lightsail uses the same generic top-level domains (TLDs) as Route 53. For a list of generic TLDs that you can use to register domains in Lightsail, see [Domains that you can register with Amazon Route 53](#) in the Amazon Route 53 Developer Guide.

If the TLD isn't included in the list, or if you would like to register a geographic domain, we recommend you use the Route 53 console. Your geographic domain will be available in the

Lightsail console after it has been registered using Route 53. For more information, see [Geographic top-level domains](#) in the Amazon Route 53 Developer Guide.

Pricing for domain registration

Lightsail uses Route 53 for domain registration. Therefore, the Route 53 pricing also applies to Lightsail registrations.

For information about the cost of registering domains, see [Domains that you can register in Amazon Route 53](#) in the Amazon Route 53 Developer Guide.

Additional information about domains

The following articles can help you manage domains in Lightsail:

- [DNS](#)
- [Format domain names](#)
- [Manage a Lightsail domain in Amazon Route 53](#)
- [Create a DNS zone to manage your domain's DNS records](#)
- [Domain registration renewal](#)
- [Edit or delete a DNS zone](#)
- [Point your domain to a load balancer](#)
- [Point your domain to a distribution](#)
- [Point your domain to an instance](#)
- [Route traffic for your domain to a container service](#)

DNS in Amazon Lightsail

People can access the web application on your Lightsail instance by browsing to the public internet protocol (IP) address of your instance, which could be an IPv4 or IPv6 address. However, IP addresses are complex and difficult for people to remember. Therefore, you should have people browse to an easy-to-remember domain name, like `example.com`, to access the web application on your instance. This is achieved through the Domain Name System (DNS), which functions as a directory that maps registered domain names to IP addresses.

To route traffic for your domain name to your Lightsail instance, you add an address (A) record that points your domain name to the static IPv4 address of your instance, or a AAAA record that points to the IPv6 address of your instance. If you registered a domain name using Lightsail, you can manage the DNS records from the DNS zone that was created when you registered the domain name. If your domain was registered through another registrar, you can manage the DNS records at the registrar or you can transfer management of your domain's DNS to Lightsail.

To make it easier to map your domain name to your Lightsail instance, we recommend that you transfer management of your domain's DNS records to Lightsail by creating a DNS zone. For more information, see [Create a DNS zone to manage your domain's DNS records](#). You can create up to six DNS zones in Lightsail. If you require more than six DNS zones, we recommend using Route 53 to manage the DNS of all your domains. You can use Route 53 to point your domain name to your Lightsail instance. For more information about managing DNS with Route 53, see [Use Amazon Route 53 to point a domain to an instance](#).

DNS terminology

So that you can manage DNS for your domain, there are terms you should be familiar with.

Apex domain / root domain

An apex domain, also known as a root domain, is a domain that does not contain a subdomain part. An example of an apex domain is `example.com`. Whereas, subdomain examples are `www.example.com` and `blog.example.com`. These are subdomains because they contain the `www` and `blog` subdomain parts respectively.

Domain Name System (DNS)

DNS routes easy-to-remember domain names, such as `example.com`, to the IP addresses of web servers.

For more information, see [Domain Name System](#) on *Wikipedia*.

DNS record

A DNS record is a mapping parameter. It tells the DNS server which IP address or hostname a domain or subdomain is associated with.

For more information, see [List of DNS record types](#) on *Wikipedia*.

DNS zone

A DNS zone is a container that holds information about how you want to route traffic on the internet for a specific domain, such as `example.com`, and its subdomains, such as `blog.example.com`.

For more information, see [DNS zone](#) on *Wikipedia*.

Domain name registrar

A domain name registrar, also known as a domain name provider, is a company or organization that manages the assignment of domain names. You can purchase a domain or manage an existing domain using Lightsail, Amazon Route 53 or any other domain name registrar.

For more information, see [Domain name registrar](#) on *Wikipedia*.

Name server

A name server routes traffic to your domain. In Lightsail, the name server is an AWS instance that runs a network service to help translate easy-to-remember domain names to IP addresses. Lightsail provides several AWS name server options (e.g., `ns-NN.awsdns-NN.com`) to route traffic to your domain. You can choose from among these AWS name servers when you change your domain using a domain registrar.

For more information, see [Name server](#) on *Wikipedia*.

Subdomain

A subdomain is anything in the domain hierarchy, other than the root domain, that is part of the larger domain. For example, `blog` is the subdomain part of the `blog.example.com` subdomain.

For more information, see [Subdomain](#) on *Wikipedia*.

Time to live (TTL)

TTL dictates the lifespan of a DNS record on local resolving name servers; for example, a shorter time means less time to wait until the changes go into effect. TTL cannot be configured in the Lightsail DNS zone. Instead, all Lightsail DNS records default to a TTL of 60 seconds.

For more information, see [Time to live](#) on *Wikipedia*.

Wildcard DNS record

A wildcard DNS record matches requests for non-existent domain names. A wildcard DNS record is specified by using the asterisk symbol (*) as the leftmost part of a domain name, such as *.example.com or *example.com.

Note

Lightsail DNS zones support wildcard records for name server domains (*awsdns.com) defined in a Name Server (NS) record.

DNS record types supported in the Lightsail DNS zone

Address (A) record

An A record maps a domain, such as example.com, or a subdomain, such as blog.example.com, to a web server's IP address.

For example, in the Lightsail DNS zone, you want to direct web traffic for example.com (the apex of the domain) to your instance. You would create an A record, enter an @ symbol into the **Subdomain** text box, and enter the IP address of your web server into the **Resolves to address** text box.

For more information about the A record, see [List of DNS record types](#) on *Wikipedia*.

AAAA record

An AAAA record maps a domain, such as example.com, or a subdomain, such as blog.example.com, to a web server's IPv6 address.

For example, in the Lightsail DNS zone, you want to direct web traffic for example.com (the apex of the domain) to your instance over the IPv6 protocol. You would create an AAAA record, enter an @ symbol into the **Subdomain** text box, and enter the IP address of your web server into the **Resolves to address** text box.

For more information about the AAAA record, see the [Domain Name System for IPv6](#) on *Wikipedia*.

Note

Lightsail does not support static IPv6 addresses. If you delete your Lightsail resource and create a new resource, or if you disable and re-enable IPv6 on the same resource, you might need to update your AAAA record to reflect the latest IPv6 address for the resource.

Canonical name (CNAME) record

A CNAME record maps an alias or subdomain, such as `blog.example.com`, to another domain or subdomain.

For example, in the Lightsail DNS zone, you want to direct web traffic for `www.example.com` to `example.com`. You would create an alias CNAME record for `www` with a "resolves to" address of `example.com`.

For more information, see [CNAME Record](#) on *Wikipedia*.

Mail exchanger (MX) record

An MX record maps a subdomain, such as `mail.example.com`, to an email server address with values for priority when multiple servers are defined.

For example, in the Lightsail DNS zone you want to direct mail for `mail.example.com` to the `10 inbound-smtp.us-west-2.amazonaws.com` Amazon WorkMail server. You would create an MX record with a subdomain of `example.com`, a priority of `10`, and a "resolves to" address of `inbound-smtp.us-west-2.amazonaws.com`.

For more information, see [MX Record](#) on *Wikipedia*.

Name server (NS) record

An NS record delegates a subdomain, such as `test.example.com`, to a name server, such as `ns-NN.awsdns-NN.com`.

For more information, see [Name server](#) on *Wikipedia*.

Service locator (SRV) record

An SRV record maps a subdomain, such as `service.example.com`, to a service address with values for priority, weight, and port number. Telephony or instant messaging are a couple of the services typically associated with SRV records.

For example, in the Lightsail DNS zone, you want to direct traffic for `service.example.com` to `1 10 5269 xmpp-server.example.com`. You would create an SRV record with a priority of 1, a weight of 10, a port number of 5269, and a "maps to" address of `xmpp-server.example.com`.

For more information, see [SRV Record](#) on *Wikipedia*.

Text (TXT) record

A TXT record maps a subdomain to plaintext. You create TXT records to confirm ownership of your domain to a service provider.

For example, in the Lightsail DNS zone, you want to respond with `23223a30-7f1d-4sx7-84fb-31bdes7csdbb` when the `_amazonchime.example.com` hostname is queried. You would create a TXT record with a subdomain value of `_amazonchime` and a "responds with" value of `23223a30-7f1d-4sx7-84fb-31bdes7csdbb`.

For more information, see [TXT Record](#) on *Wikipedia*.

Topics

- [Create a Lightsail DNS zone to manage your domain's DNS records](#)
- [Edit or delete a Lightsail DNS zone](#)
- [How internet traffic is routed to your website in Lightsail](#)
- [Point your Lightsail domain to an instance](#)
- [Point your Lightsail domain to a load balancer](#)
- [Update your Lightsail domain name servers to use another DNS service](#)
- [Use Amazon Route 53 to point a domain to a Lightsail instance](#)

Create a Lightsail DNS zone to manage your domain's DNS records

To route traffic for a domain name, such as `example.com`, to an Amazon Lightsail instance, you add a record to the Domain Name System (DNS) of your domain. You can manage the DNS records of your domain using the registrar where you registered your domain, or you can manage them using Lightsail.

We recommend that you transfer management of your domain's DNS records to Lightsail. This allows you to efficiently administer your domain and compute resources together in one place—Lightsail. You can manage the DNS records of your domain using Lightsail by creating a Lightsail

DNS zone. You can create up to six Lightsail DNS zones. If you require more than six DNS zones, because you manage more than six domain names, we recommend using Amazon Route 53 to manage the DNS of all of your domains. You can use Route 53 to route traffic for your domain to your Lightsail resources. For more information about managing DNS with Route 53, see [Use Amazon Route 53 to point a domain to an instance](#).

This guide shows you how to create a Lightsail DNS zone for your domain, and how to transfer management of your domain's DNS records to Lightsail. After transferring management of your domain's DNS records to Lightsail, you will continue to manage renewals and billing for your domain at your domain's registrar.

Important

Any changes you make to the DNS of your domain might require several hours to propagate through the internet's DNS. Because of this, you should keep the DNS records of your domain in place at your domain's current DNS hosting provider while the transfer of management to Lightsail propagates. This ensures that traffic for your domain continues to route to your resources uninterrupted while the transfer takes place.

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Create a DNS zone in the Lightsail console](#)
- [Step 3: Add records to the DNS zone](#)
- [Step 4: Change the name servers at your domain's current DNS hosting provider](#)

Step 1: Complete the prerequisites

Complete the following prerequisites if you haven't already done so:

1. Register a domain name. Then, confirm that you have administrative access to edit the domain's name servers.

If you need a registered domain name, you can register a domain using Lightsail. For more information, see [Domain registration](#).

2. Confirm that the necessary DNS record types for your domain are supported by the Lightsail DNS zone. The Lightsail DNS zone currently supports address (A and AAAA), canonical name

(CNAME), mail exchanger (MX), name server (NS), service locator (SRV), and text (TXT) record types. For NS records, you can use wildcard DNS record entries.

If the DNS record types required for your domain are not supported by the Lightsail DNS zone, you might want to use Route 53 as your domain's DNS hosting provider because it supports a greater number of record types. For more information, see [Supported DNS Record Types](#) and [Making Amazon Route 53 the DNS Service for an Existing Domain](#) in the Amazon Route 53 Developer Guide.

3. Create a Lightsail instance to which you will point your domain. For more information, see [Create an instance](#).
4. Create a static IP and attach it to your Lightsail instance. For more information, see [Create a static IP and attach it to an instance](#).

Step 2: Create a DNS zone in the Lightsail console

Complete the following steps to create a DNS zone in Lightsail. When you create a DNS zone, you must specify the domain name that the DNS zone will apply to.

1. Sign in to the [Lightsail console](#).
2. Choose the **Domains & DNS** tab, and then choose **Create DNS zone**.
3. Choose one of the following options:
 - **Use a domain that is registered with Amazon Route 53**, to specify a domain that was registered with Amazon Route 53
 - **Use a domain from another registrar**, to specify a domain that was registered using another registrar
4. Select or enter your registered domain name, such as `example.com`.

It isn't necessary to include `www` when entering your domain name. You can add the `www` using an address (A) record as part of the [Step 3: Add records to the DNS zone](#) section later in this guide.

Note

Lightsail DNS zones are created in the Virginia (us-east-1) AWS Region. You will get a resource name conflict error ("some names are already in use") if you named a resource in that Region the same as the Lightsail DNS zone (`example.com`) you want to create.

To resolve the error, [create a snapshot of the resource](#). [Create a new resource from the snapshot](#) and give it a new, unique name. Then, delete the original resource that is named the same as the domain for which you want to create a Lightsail DNS zone.

5. Choose **Create DNS zone**.


You are redirected to the DNS zone **Assignments** page, where you can manage domain resource assignments. Use assignments to point a domain to your Lightsail resources, such as load balancers and instances.

Step 3: Add records to the DNS zone

Complete the following steps to add records to your domain's DNS zone. DNS records specify how internet traffic is routed for the domain. For example, you could route traffic for the apex of your domain, such as `example.com`, to one instance, and route traffic for a subdomain, such as `blog.example.com`, to a different instance.

1. From the DNS zone assignments page, choose the **DNS records** tab.

Your DNS zones are listed in the **Domains & DNS** tab of the [Lightsail console](#).

 **Note**

On the DNS zone **Assignments** page, you can add, remove, or change which Lightsail resource your domain points to. You can point domains at Lightsail instances, distributions, container services, load balancers, static IP addresses and more. On the **DNS records** page, you can add, edit, or delete your domain's DNS records.

2. Choose one of the following record types:

Address (A) record

An A record maps a domain, such as `example.com`, or a subdomain, such as `blog.example.com`, to a web server's or instance's IPv4 address, such as `192.0.2.255`.

1. In the **Record name** text box, enter the target subdomain for the record, or enter an @ symbol to define the apex of your domain.


2. In the **Resolves to** text box, enter the target IP address for the record, select your running instance, or configured load balancer. When you select a running instance, the public IP address of that instance is automatically added.
3. Select **Is AWS resource alias** to route traffic to your Lightsail and AWS resources, such as a distribution or container service. You can also route traffic from one record in a DNS zone to another record.

 **Note**

We recommend that you attach a static IP to your Lightsail instance and then choose the static IP as the value that the record resolves to. For more information, see [Create a static IP](#).

AAAA record

An AAAA record maps a domain, such as `example.com`, or a subdomain, such as `blog.example.com`, to a web server's or instance's IPv6 address, such as `2001:0db8:85a3:0000:0000:8a2e:0370:7334`.

 **Note**

Lightsail does not support static IPv6 addresses. If you delete your Lightsail resource and create a new resource, or if you disable and re-enable IPv6 on the same resource, you might need to update your AAAA record to reflect the latest IPv6 address for the resource.

1. In the **Record name** text box, enter the target subdomain for the record, or enter an @ symbol to define the apex of your domain.
2. In the **Resolves to** text box, enter the target IPv6 address for the record, select your running instance, or configured load balancer. When you select a running instance, the public IPv6 address of that instance is automatically added.
3. Select **Is AWS resource alias** to route traffic to your Lightsail and AWS resources, such as a distribution or container service. You can also route traffic from one record in a DNS zone to another record.

Canonical name (CNAME) record

A CNAME record maps an alias or subdomain, such as `www.example.com`, to another domain, such as `example.com`, or another subdomain, such as `blog.example.com`.

1. In the **Record name** text box, enter the subdomain for the record.
2. In the **Route traffic to** text box, enter the target domain or subdomain for the record.

Mail exchanger (MX) record

An MX record maps a subdomain, such as `mail.example.com`, to an email server address with priority values when multiple servers are defined.

1. In the **Record name** text box, enter the subdomain for the record.
2. In the **Priority** text box, enter the priority for the record. This is important when adding records for multiple servers.
3. In the **Route traffic to** text box, enter the target domain or subdomain for the record.

Service locator (SRV) record

An SRV record maps a subdomain, such as `service.example.com`, to a service address with values for priority, weight, and port number. Telephony or instant messaging are a couple of the services typically associated with SRV records.

1. In the **Record name** text box, enter the subdomain for the record.
2. In the **Priority** text box, enter the priority for the record.
3. In the **Weight** text box, enter a relative weight for SRV records with the same priority.
4. In the **Route traffic to** text box, enter the target domain or subdomain for the record.
5. In the **Port** text box, enter the port number in which a connection to the service can be made.

Text (TXT) record

A TXT record maps a subdomain to plain text. You create TXT records to confirm ownership of your domain to a service provider.

1. In the **Record name** text box, enter the subdomain for the record.
2. In the **Responds with** text box, enter the text response that is given when the subdomain is queried.

Note

The input text doesn't need to be enclosed with quotes.

3. When you're done adding the record, choose the **Save** icon to save your changes.

The record is added to the DNS zone. Repeat the above steps to add multiple records to your domain's DNS zone.

Note

Time to live (TTL) for DNS records cannot be configured in the Lightsail DNS zone. Instead, all Lightsail DNS records default to a TTL of 60 seconds. For more information, see [Time to live](#) on Wikipedia.

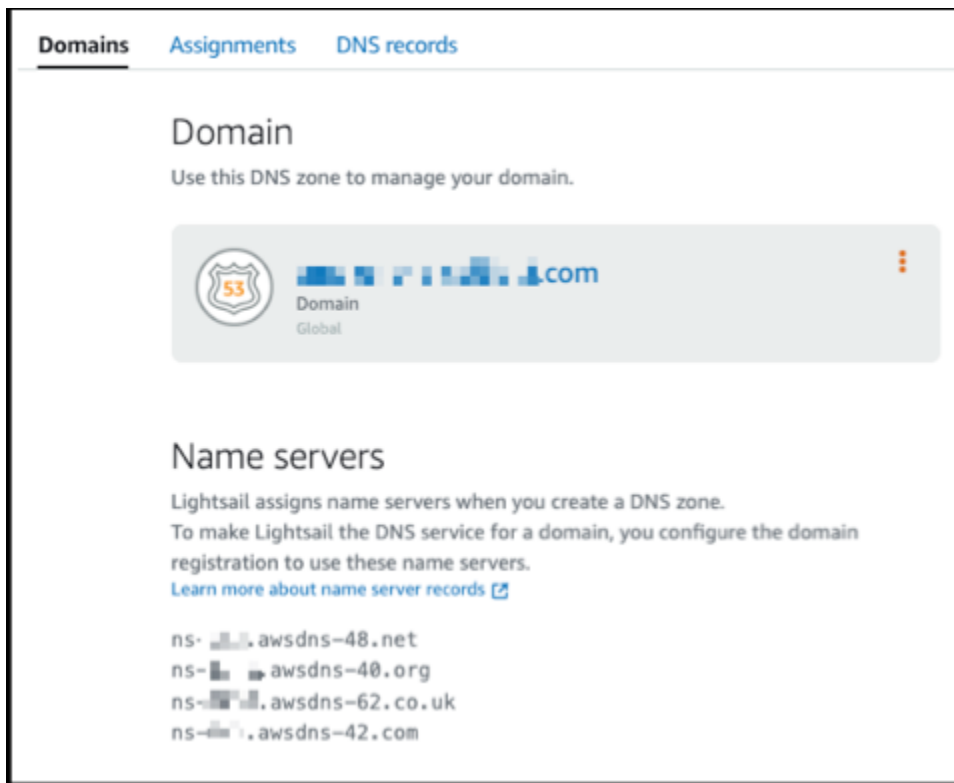
Step 4: Change the name servers at your domain's current DNS hosting provider

Complete the following steps to transfer management of your domain's DNS records to Lightsail. To do this, you sign in to the website of your domain's current DNS hosting provider, and change your domain's name servers to the Lightsail name servers.

Important

If web traffic is currently being routed to your domain, make sure that all of the existing DNS records are present in the Lightsail DNS zone before changing the name servers at your domain's current DNS hosting provider. This way, traffic continually flows uninterrupted after the transfer to the Lightsail DNS zone.

1. Write down the Lightsail name servers that are listed on your domain's DNS zone management page. The name servers are located on the **Domains** tab of your Lightsail DNS zone.



2. Sign in to your domain's current DNS hosting provider's website.
3. Find the page where you can edit your domain's name servers.

For more information about locating this page, see the documentation from your domain's current DNS hosting provider.

4. Enter the Lightsail name servers, and remove other name servers listed.
5. Save your changes.

Allow time for the name server change to propagate through the internet's DNS, which might take several hours. After that is completed, internet traffic for your domain should begin routing through the Lightsail DNS zone.

Next steps

- [Edit or delete a DNS zone](#)
- [Create a load balancer and attach instances to it](#)

Edit or delete a Lightsail DNS zone

You can add, edit, or delete the DNS records in your domain's DNS zone. You can also delete your domain's DNS zone in Amazon Lightsail if you want to transfer management of your domain's DNS records to another DNS hosting provider or back to the registrar where you registered your domain.

Note

Before you can edit records using the DNS editor in the Lightsail console, you must transfer management of your domain's DNS records to Lightsail. For more information, see [Create a DNS zone to manage your domain's DNS records](#).

Edit DNS records

You can edit the DNS records for your domain's DNS zone at any time using the Lightsail console.

To edit the DNS zone

1. Sign in to the Lightsail console.
2. Choose the **Domains & DNS** tab, and then choose the name of the DNS zone you want to edit.
3. On the DNS zone **DNS records** page, choose one of the following options:
 - To add a new record, choose **Add record**.
 - To edit an existing record, choose the **Edit** icon next to the record you want to edit.
 - To delete an existing record, choose the **Delete** icon next to the record you want to delete.
4. When you're done, choose the **Save** icon to save your changes.

Note

Allow time for the DNS record changes to propagate through the internet's DNS, which may take several hours.

Delete a DNS zone

You can delete your domain's DNS zone in Lightsail.

Important

If you plan to continue routing traffic through your domain, prepare a different DNS hosting provider before deleting your domain's DNS zone in Lightsail. Otherwise, all traffic to your website stops when you delete the Lightsail DNS zone.

To delete a DNS zone

1. On the Lightsail console home page, choose the **Domains & DNS** tab.
2. Choose the name of the DNS zone you want to delete.
3. Choose the vertical ellipsis menu (:). Then, choose the **Delete** option.
4. Choose **Delete DNS zone** to confirm the deletion.

The DNS zone is deleted from Lightsail.

How internet traffic is routed to your website in Lightsail

All computers on the internet, including smart phones, laptops, and website servers, communicate with one another by using unique strings of characters. These strings, known as IP addresses, are in one of the following formats:

- Internet Protocol version 4 (IPv4) format, such as 192.0.2.44
- Internet Protocol version 6 (IPv6) format, such as 2001:DB8::/32

When you open a browser and go to a website, you don't have to remember and enter a long string of characters like that. Instead, you can enter a domain name like **example.com** and still end up in the right place. This is achieved through the Domain Name System (DNS), which functions as a directory that maps registered domain names to IP addresses.

Contents

- [Overview of how you configure Lightsail to route internet traffic for your domain](#)
- [How traffic is routed for your domain](#)
- [Next steps](#)

Overview of how you configure Lightsail to route internet traffic for your domain

This overview explains how to use Lightsail to register and configure a domain that routes internet traffic to your website or web application.

1. Register your domain name. For an overview, see [Domain registration](#).
2. After you register your domain name, Lightsail automatically creates a DNS zone that has the same name as the domain.
3. The Lightsail console allows you to easily assign a domain to a Lightsail resource, such as an instance or load balancer. You can also create DNS records in your DNS zone to route traffic to your resources. Each record includes information about how you want to route traffic for your domain, such as the following:

Name

The name of the record corresponds with the domain name (example.com) or subdomain name (www.example.com, retail.example.com). The name of every record in a DNS zone must end with the name of the DNS zone. For example, if the name of the DNS zone is example.com, all record names must end in example.com.

Type

The record type usually depends on the type of resource that you want traffic to be routed to. For example, to route traffic to an email server, you specify **MX** for **Type**. To route traffic for your domain name to your Lightsail instance, you add an **A** record that points your domain name to the static IPv4 address of your instance, or a **AAAA** record that points to the IPv6 address of your instance.

4. Target

The target is where you want traffic to be routed to. You can create alias records that route traffic to Lightsail instances, Lightsail container services, and other Lightsail resources. For more information, see [DNS](#).

How traffic is routed for your domain

After you configure Lightsail to route your internet traffic to your resources, such as instances, load balancers, distributions, or container services, here's what happens when someone requests content for **www.example.com**.

1. A user opens a web browser, enters **www.example.com** in the address bar, and presses **Enter**.
2. The request for **www.example.com** is routed to a DNS resolver, which is typically managed by the user's internet service provider (ISP). ISPs can be cable internet providers, DSL broadband providers, or corporate networks.
3. The DNS resolver for the ISP forwards the request for **www.example.com** to a DNS root name server.
4. The DNS resolver forwards the request for **www.example.com** again, this time to one of the TLD name servers for **.com** domains. The name server for **.com** domains responds to the request with the names of the four name servers that are associated with the **example.com** domain.

The DNS resolver caches (stores) the four name servers. The next time someone browses to **example.com**, the resolver skips steps 3 and 4 because it already has the name servers for **example.com**. The name servers are typically cached for two days.

5. The DNS resolver chooses a name server and forwards the request for **www.example.com** to that name server.
6. The name server looks in the **example.com** DNS zone for the **www.example.com** record and gets the associated value, such as the IP address for a web server (192.0.2.44). Then, the name server returns the IP address to the DNS resolver.
7. The DNS resolver finally has the IP address that the user needs. The resolver returns that value to the web browser.
8. The web browser sends a request for **www.example.com** to the IP address that it got from the DNS resolver. This is where your content is, for example, a web server running on an Lightsail instance or container service that's configured as a website endpoint.
9. The web server or other resource at 192.0.2.44 returns the web-page for **www.example.com** to the web browser, and the web browser displays the page.

Next steps

- [DNS](#)
- [Point your domain to an instance](#)
- [Point your domain to a load balancer](#)
- [Point your domain to a distribution](#)

Point your Lightsail domain to an instance

You can use the DNS zone in Amazon Lightsail to point a registered domain name, like **example.com**, to your website running on a Lightsail instance, also known as a virtual private server (VPS). You can create up to six DNS zones in your Lightsail account. Not all DNS record types are supported. For more information about Lightsail DNS zones, see [DNS](#).

If you expect to create more than six DNS zones or use DNS record types that aren't supported in Lightsail, we recommend using an Amazon Route 53 hosted zone. With Route 53, you can manage the DNS for up to 500 domains. It also supports a greater variety of DNS record types. For more information, see [Working with hosted zones](#) in the Amazon Route 53 Developer Guide.

This guide shows you how to edit the DNS records for a domain managed in Lightsail so that it points to your Lightsail instance. Allow up to 48 hours for any DNS zone changes to propagate through the internet's DNS.

Prerequisites

Complete the following prerequisites if you haven't already done so:

- Register a domain name using Lightsail. For more information, see [Register a new domain](#).
- If you already registered a domain but you're not using Lightsail to manage its records, then you must transfer management of the DNS records for your domain to Lightsail. For more information, see [Create a DNS zone to manage your domain's DNS records](#).
- The default dynamic public IP address attached to your Lightsail instance changes every time you stop and restart the instance. Create a static IP and attach it to your instance to keep the public IP address from changing. In this guide, you create a DNS record in your domain's DNS zone that resolves to the static IP address so you don't have to update your domain's DNS records every time you stop and restart your instance. For more information, see [Create a static IP and attach it to an instance](#).

Optional—You can leave IPv6 enabled for your Lightsail instance. The IPv6 address persists when you stop and start your instance. For more information, see [Enable and disable IPv6](#).

Assign a domain to a Lightsail instance

Use one of the following methods to assign a domain to an instance in Lightsail:

- [Instance domains tab](#)

- [Static IP domains tab](#)
- [DNS zone assignments tab](#)

Instance domains tab

Complete the following procedure to assign your domain to a Lightsail instance in the instance **Domains** tab of the Lightsail console.

To assign your domain by using the instance Domains tab

1. Sign in to the [Lightsail console](#).
2. Choose the instance name that you want to assign the domain to.
3. Choose **Assign domain** in the **Domains** tab.
4. Select the domain that you want to assign to your Lightsail instance.
5. Verify that the routing information is correct, and then choose **Assign**.

Optional

To edit or remove your domain assignment from the instance, choose the edit icon or the waste bin icon next to the domain name.

Static IP domains tab

Complete the following procedure to assign your domain to a Lightsail instance in the static IP **Domains** tab of the Lightsail console.

To assign your domain by using the static IP Domains tab

1. Sign in to the [Lightsail console](#).
2. Choose the **Networking** tab.
3. Choose the static IP that you want to assign the domain to.
4. Choose **Assign domain** in the **Domains** tab.
5. Select the domain that you want to assign to your static IP.
6. Verify that the routing information is correct, and then choose **Assign**.

Optional

To edit or remove your domain assignment from the static IP, choose the edit icon or the waste bin icon next to the domain name.

DNS zone assignments tab

Complete the following procedure to assign your domain to a Lightsail instance in the **Assignments** tab of the DNS zone.

To assign your domain by using the Assignments tab

1. Sign in to the [Lightsail console](#).
2. Choose the **Domains & DNS** tab.
3. Choose the DNS zone for the domain name that you want to use.
4. Choose **Add assignment** in the **Assignments** tab.
5. Select the domain name that you want to assign to your Lightsail instance. If a static IP isn't already attached to the instance, you are prompted to attach one.
6. Verify that the routing information is correct, and then choose **Assign**.

Optional

To edit or remove your domain assignment from the resource, choose the edit icon or the waste bin icon next to the domain name.

Point your Lightsail domain to a load balancer

After you [verify that you control the domain where you want to have encrypted \(HTTPS\) traffic](#), you need to add an address (A) record to your domain's DNS hosting provider that points your domain to your Lightsail load balancer. In this guide, we show you how to add the A record to a Lightsail DNS zone, and an Amazon Route 53 hosted zone.

Add an A record using the DNS zone - Assignments page

1. On the Lightsail home page, choose **Domains & DNS**.
2. Choose the DNS zone you want to manage.
3. Choose the **Assignments** tab.
4. Choose **Add assignment**.
5. In the **Select a domain name** field, choose whether to use the domain name, or a subdomain of the domain.

6. In the **Select a resource** drop down, select the load balancer you want to assign the domain to.
7. Choose **Assign**.

Allow time for the change to propagate through the internet's DNS. This may take a few minutes to several hours.

Add an A record using the DNS zone - DNS records page

1. On the Lightsail home page, choose **Domains & DNS**.
2. Choose the DNS zone you want to manage.
3. Choose the **DNS records** tab.
4. Complete one of the following steps depending on the current state of your DNS zone:
 - If you haven't added an A record, choose **Add record**.
 - If you previously added an A record, choose the edit icon next to the existing A record listed on the page, and then skip to step 5 of this procedure.
5. Choose **A record** in the **Record type** dropdown menu.
6. In the **Record name** text box, enter one of the following options:
 - Enter @ to route traffic for the apex of your domain (e.g., `example.com`) to your load balancer.
 - Enter `www` to route traffic for the `www` subdomain (e.g., `www.example.com`) to your load balancer.
7. In the **Resolves to** text box, choose the name of your Lightsail load balancer.
8. Choose the **Save** icon.

Allow time for the change to propagate through the internet's DNS. This may take a few minutes to several hours.

Add an A record in Route 53

1. Sign in to the [Route 53 console](#).
2. In the navigation pane, choose **Hosted zones**.
3. Choose the hosted zone for the domain name that you want to use to route traffic to your load balancer.

4. Choose **Create record**.

The **Quick create record** page appears.

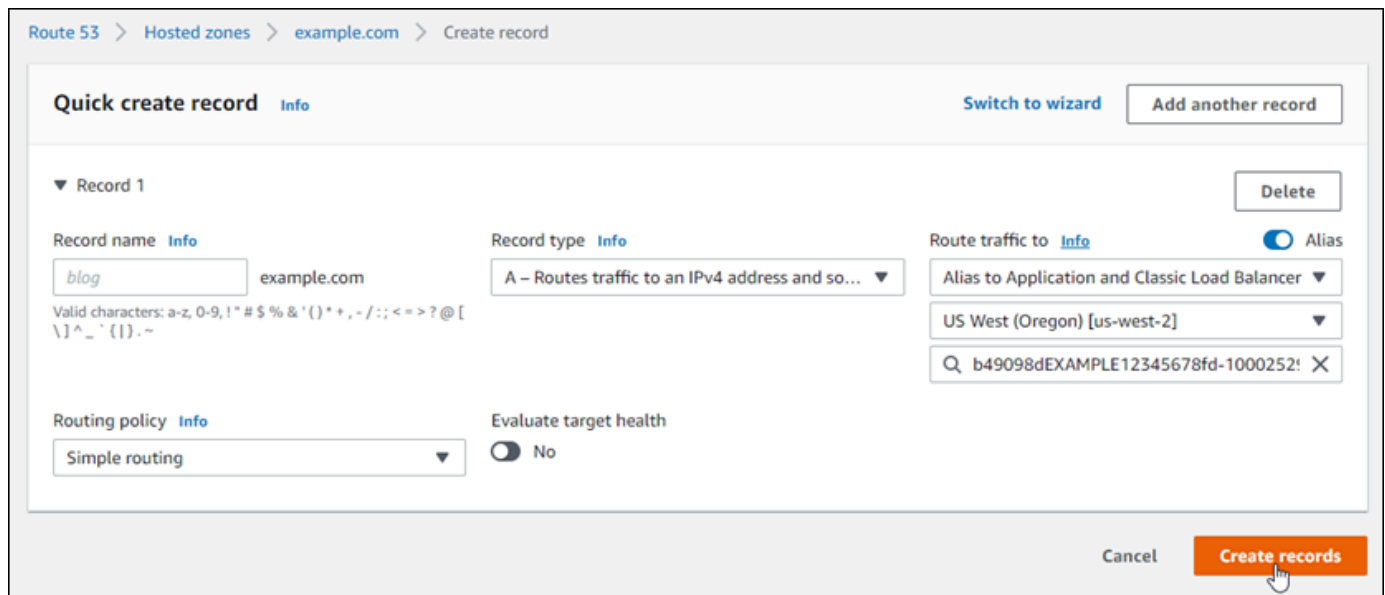
Note

If you see the **Choose routing policy** page, then choose **Switch to quick create** to switch to the quick create wizard before continuing with the following steps.

5. For **Record name**, type `www` if you plan to use the `www` subdomain (i.e., `www.example.com`) or leave it blank if you plan to use the apex of the domain (i.e., `example.com`).
6. For **Record type**, choose **A - Routes traffic to an IPv4 address and some AWS resources**.
7. Choose the **Alias** toggle to enable alias records.
8. Choose the following options for **Route traffic to**:
 - a. For **Choose endpoint**, choose **Alias to Application and Classic Load Balancer**.
 - b. For **Choose Region**, choose the AWS Region in which you created your Lightsail load balancer.
 - c. For **Choose load balancer**, enter or paste the endpoint URL (i.e., DNS name) of your Lightsail load balancer.
9. For **Routing Policy**, choose **Simple routing**, and disable the **Evaluate target health** toggle.

Lightsail already performs health checks on your load balancer. For more information, see [Health checks for your load balancer](#).

Your record should look like the following example.



The screenshot shows the 'Create record' interface in the AWS Management Console. The breadcrumb navigation at the top reads 'Route 53 > Hosted zones > example.com > Create record'. The main heading is 'Quick create record' with an 'Info' link. There are two buttons: 'Switch to wizard' and 'Add another record'. Below this, there is a section for 'Record 1' with a 'Delete' button. The form includes several fields: 'Record name' (containing 'blog') and 'Record type' (set to 'A - Routes traffic to an IPv4 address and so...'). The 'Route traffic to' section is set to 'Alias' and includes 'Alias to Application and Classic Load Balancer', 'US West (Oregon) [us-west-2]', and a search bar containing 'b49098dEXAMPLE12345678fd-1000252!'. The 'Routing policy' is set to 'Simple routing' and 'Evaluate target health' is turned off. At the bottom right, there are 'Cancel' and 'Create records' buttons.

10. Choose **Create records** to add the record to your hosted zone.

Note

Allow time for the change to propagate through the internet's DNS. This may take a few minutes to several hours.

Update your Lightsail domain name servers to use another DNS service

You can use an Amazon Lightsail DNS zone to manage the DNS records for a domain that you registered using Lightsail. Or, if you'd like, you can transfer management of DNS records for the domain to another DNS hosting provider. In this guide, we show you how to transfer management of DNS records for a domain you registered with Lightsail to another DNS hosting provider.

Important

Any changes you make to the DNS of your domain might require several hours to propagate through the internet's DNS. Because of this, you should keep the DNS records of your domain in place at your current DNS hosting provider until the transfer of

management is done. This ensures that traffic for your domain continues to route to your resources uninterrupted while the transfer takes place.

Contents

- [Complete the prerequisites](#)
- [Add records to the DNS zone](#)

Complete the prerequisites

Complete the following prerequisites if you haven't already done so:

1. Register a domain name. You can register a domain name using Lightsail. For more information, see [Register a new domain](#).
2. Use the process that's provided by your DNS service to get the name servers for your domain.

Add records to the DNS zone

Complete the following procedure to add the name servers for another DNS hosting provider into your registered domain in Lightsail.

1. Sign in to the [Lightsail console](#).
2. Choose the **Domains & DNS** tab.
3. Choose the name of the domain that you want to configure to use another DNS service.
4. Choose **Edit Name Servers**.
5. Change the names of the name servers to the name servers that you got from your DNS service when you completed the prerequisites.
6. Choose **Save**.

Use Amazon Route 53 to point a domain to a Lightsail instance

The DNS zone in Amazon Lightsail makes it easy to point a registered domain name, like example.com, to your website running on a Lightsail instance. You can create up to six Lightsail DNS zones, and not all DNS record types are supported. For more information about Lightsail DNS zones, see [DNS](#).

If the Lightsail DNS zone is too limited for you, then we recommend using an Amazon Route 53 hosted zone to manage your domain's DNS records. You can manage the DNS for up to 500 domains using Route 53, and it supports a greater variety of DNS record types. Or, you might already be using Route 53 to manage your domain's DNS records and prefer to continue using it. This guide shows you how to edit the DNS records for a domain managed in Route 53 to point to your Lightsail instance.

Prerequisites

Complete the following prerequisites if you haven't already done so:

- Register a domain name using Route 53. For more information, see [Registering a New Domain](#) in the Route 53 documentation.
- If you already registered a domain but you're not using Route 53 to manage its records, then you must transfer management of the DNS records for your domain to Route 53. For more information, see [Making Amazon Route 53 the DNS Service for an Existing Domain](#) in the Route 53 documentation.
- Create a public hosted zone for your domain in Route 53. For more information, see [Creating a Public Hosted Zone](#) in the Route 53 documentation.
- Create a static IP and attach it to your Lightsail instance. In this guide, you create a DNS record in your domain's Route 53 hosted zone that resolves to the static IP address (public IP address) of your instance. For more information, see [Create a static IP and attach it to an instance](#).

Point a domain to a Lightsail instance using Route 53

Complete the following steps to configure the two most common DNS records, address and canonical name, in Route 53 to point your domain to a Lightsail instance.

Note

This procedure is also documented in the Route 53 Developer Guide. For more information, see [Creating Records by Using the Amazon Route 53 Console](#) in the Route 53 documentation.

1. Sign in to the [Route 53 console](#).
2. In the navigation pane, choose **Hosted zones**.

- Choose the hosted zone for the domain name that you want to use to route traffic to your load balancer.
- Choose **Create record**.

The **Quick create record** page appears.

Note

If you see the **Choose routing policy** page, then choose **Switch to quick create** to switch to the quick create wizard before continuing with the following steps.

- For **Record type**, choose one of the following options:

A - Routes traffic to an IPv4 address and some AWS resources

An address (A) record maps a domain, such as `example.com`, or a subdomain, such as `blog.example.com`, to a web server's IP address, such as `192.0.2.255`.

- Keep the **Record name** text box empty to point the apex of your domain, such as `example.com`, to an IP address, or enter a subdomain.
- Choose **A - Routes traffic to an IPv4 address and some AWS resources** in the **Record type** drop-down menu.
- Enter the static IP address (public IP address) of your Lightsail instance in the **Value** text box.

4. Keep the TTL of 300, and the routing policy as **Simple routing**.

Route 53 > Hosted zones > example.com > Create record

Quick create record Info Switch to wizard Add another record

▼ Record 1 Delete

Record name Info example.com Record type Info A - Routes traffic to an IPv4 address and so... Value Info Alias

Valid characters: a-z, 0-9, !*# \$ % & ' () * + , - / : ; < = > ? @ [\] ^ _ ` { } . ~
Enter multiple values on separate lines.

TTL (seconds) Info Routing policy Info Simple routing

1m 1h 1d
Recommended values: 60 to 172800 (two days)

Cancel Create records

CNAME - Routes traffic to another domain name and to some AWS resources

A canonical name (CNAME) record maps an alias or subdomain, such as `www.example.com`, to a domain, such as `example.com`, or a subdomain, such as `www2.example.com`. A CNAME record redirects one domain to another.

1. Enter a subdomain in the **Record name** text box.
2. Choose **CNAME - Routes traffic to another domain name and to some AWS resources** in the **Record type** drop-down menu.
3. Enter a domain (i.e., `example.com`) or subdomain (i.e., `another.example.com`) in the **Value** text box.
4. Keep the TTL of 300, and the routing policy as **Simple routing**.

Route 53 > Hosted zones > example.com > Create record

Quick create record Info Switch to wizard Add another record

▼ Record 1 Delete

Record name Info example.com Record type Info Value Info Alias

Valid characters: a-z, 0-9, !*#\$%&'()*+,-./:;<=>?@[\]^_`{|}~.~
Enter multiple values on separate lines.

TTL (seconds) Info Routing policy Info

Recommended values: 60 to 172800 (two days)

Cancel Create records

6. Choose **Create records** to add the record to your hosted zone.

Note

Allow time for the change to propagate through the internet's DNS. This may take a few minutes to several hours.

To edit an existing record set in the Route 53 hosted zone, choose the record to edit, enter your changes, and then choose **Save**.

Register a new domain in Lightsail

You can register new domains using Amazon Lightsail. Lightsail domains are registered through Amazon Route 53, a highly available and scalable DNS web service. If you have domains that are registered with other providers, you can transfer DNS management of those domains to Lightsail. You can also point those domains to your Lightsail resources.

Choose one of the following procedures to register a new domain with Lightsail:

- For registering a new domain, see [Register a new domain by using Lightsail](#).
- For an existing domain, see [Create a DNS zone to manage your domain's DNS records](#).
- For moving a domain to another registrar, see [Manage a Lightsail domain in Amazon Route 53](#).

Before you start, note the following considerations for domain registration:

Domain registration pricing

For information about the cost to register domains, see [Amazon Route 53 pricing guide](#).

Domain service quotas

There is a limit for how many domains you can register. For more information, see [Service quotas](#) in the Amazon Route 53 Developer Guide. Contact Route 53 if you want to increase the limit.

Supported domains

Lightsail supports the registration of all generic top-level domains (TLDs). For a list of supported TLDs, see [Domains that you can register with Amazon Route 53](#) in the Amazon Route 53 Developer Guide.

You must use Route 53 to register geographic top-level domains. For more information, see [Geographic top-level domains](#) in the Amazon Route 53 Developer Guide.

Domain names can't be changed after registration

If you accidentally register the wrong domain name, you won't be able to change it. Instead, you must register another domain name and specify the correct name. There are no refunds for accidentally registered domain names.

Charges for DNS zones

When you register a domain with Lightsail, we automatically create a DNS zone for the domain. Lightsail does not charge a fee for the DNS zone.

Register a new domain by using Lightsail

Contents

- [Complete the prerequisites](#)
- [Register a new domain](#)
- [Verify the domain contact information](#)

Complete the prerequisites

Complete the following prerequisites if you haven't already done so:

1. Confirm that the necessary DNS record types for your domain are supported by the Lightsail DNS zone. The Lightsail DNS zone currently supports address (A), canonical name (CNAME), mail exchanger (MX), name server (NS), service locator (SRV), and text (TXT) record types. For NS records, you can use wildcard DNS record entries.

If the DNS record types required for your domain are not supported by the Lightsail DNS zone, you might want to use Route 53 as your domain's DNS hosting provider. Route 53 supports more record types. For more information, see [Supported DNS Record Types](#) and [Making Amazon Route 53 the DNS Service for an Existing Domain](#) in the Amazon Route 53 Developer Guide.

Register a new domain

To register a new domain

1. Sign in to the [Lightsail console](#).
2. Choose the **Domains & DNS** tab.
3. Choose **Register domain**, and specify the domain that you want to register.
 - a. Enter the domain name that you want to register, and choose **Check availability** to find out whether the domain name is available. If the domain is available, continue to **Automatic domain renewal**.
 - b. If the domain name isn't available, you see a list of other domains that you might want to register instead of your first choice or in addition to your first choice. Choose **Select** for the domain that you want to register.
4. Choose whether to automatically renew your domain registration before the expiration date. When you register a domain name, you own it for a year by default. If you don't renew your domain name registration, it expires and someone else can register the domain name. To make sure that you keep your domain name, you can choose to renew it automatically every year, or select a longer term.
5. In the **Domain contact information** section, enter contact information for the domain registrant, administrator, and technical contacts. For more information, see [Values that you specify when you register or transfer a domain](#).

Note the following considerations:

First name and Last name

For **First name** and **Last name**, we recommend that you specify the name on your official ID. For some changes to domain settings, some domain registries require that you provide proof of identity. The name on your ID must match the name of the registrant contact for the domain.

Different contacts

By default, we use the same information for all three contacts. If you want to enter different information for one or more contacts, uncheck the **Same as registrant** checkbox and enter the new contact information.

6. In the **Privacy protection** section, choose whether you want to hide your contact information from WHOIS queries.

For more information, see the following topics:

- [Privacy protection](#)
- [Domains that you can register with Amazon Route 53](#)

7. Choose **Register domain** to continue. The **DNS zones** and **Summary** sections show information about the domain's DNS zone, pricing, and renewal schedule.
8. You must accept the [Amazon Route 53 domain name registration agreement](#) before you can register your domain.

Verify the domain contact information

After you register your domain, you must verify that the email address for the registrant contact is valid.

We automatically send a verification email from one of the following email addresses:

noreply@registrar.amazon.com

For domains with Amazon Registrar as the registrar

noreply@domainnameverification.net

For domains with our registrar associate, Gandi, as the registrar. To determine who the registrar is for your TLD, see [Domains that you can register with Amazon Route 53](#) in the Amazon Route 53 Developer Guide.

Use the following procedure to complete the domain verification process.

To complete domain verification

1. When you receive the verification email, choose the link in the email that verifies that the email address is valid. If you don't receive the email immediately, check your junk email folder.
2. Return to the Lightsail console. If the status doesn't automatically update to **Verified**, choose **Refresh status**.

Important

The registrant contact must follow the instructions in the email to verify that the email was received, or we will suspend the domain as required by ICANN. When a domain is suspended, it's not accessible on the internet.

3. When domain registration is complete, choose whether to use Lightsail as your DNS service, or use a different DNS service.

- **Lightsail**

In the DNS zone that Lightsail created when you registered the domain, create records to tell Lightsail how you want to route traffic for the domain and subdomains.

For example, when someone enters your domain name in a browser and that query is forwarded to Lightsail, do you want Lightsail to respond to the query with the IP address of a web server or with the name of a load balancer? For more information, see [Edit or delete a DNS zone](#).

- **Using another DNS service**

Configure your new domain to route DNS queries to a DNS service other than Lightsail. For more information, see [Update the name servers for your domain when you want to use another DNS service](#).

View information about domains that are registered with Amazon Registrar

You can view information about .com, .net, and .org domains that were registered using Amazon Lightsail and Amazon Route 53, for which Amazon Registrar is the registrar. This information

includes details such as when the domain was originally registered and contact information for the domain owner and for the technical and administrative contacts.

Note the following:

Email domain contacts when privacy protection is active

If privacy protection is active for the domain, contact information for the registrant, technical, and administrative contacts is replaced with contact information for the Amazon Registrar privacy service. For example, if the **example.com** domain is registered with Amazon Registrar and if privacy protection is active, the value of **Registrant Email** in the response to a WHOIS query would be similar to `owner1234@example.com.whoisprivacyservice.org`.

To contact one or more domain contacts when privacy protection is active, send an email to the corresponding email addresses. We'll automatically forward your email to the applicable contact.

Report abuse

To report any illegal activity or violation of the [Acceptable Use Policy](#), including inappropriate content, phishing, malware, or spam, send an email to **abuse@amazon.com**.

To view information about domains that are registered with Amazon Registrar

1. In a web browser, go to one of the following websites. Both websites display the same information. However, they use different protocols and display the information in different formats:
 - **WHOIS:** <https://registrar.amazon.com/whois>
 - **RDAP:** <https://registrar.amazon.com/rdap>
2. Enter the name of the domain that you want to view information about, and choose **Search**. If the domain you search for was not registered using Amazon Lightsail or Route 53, then you will see a message stating that the domain is not in the registrar database.

Format domain names in Lightsail

To help people access the website or application, choose a domain name that's easy to remember. Domain names (and the names of DNS zones, and records) consist of a series of labels separated by periods (.). Naming requirements depend on whether you're registering a domain name or specifying the name of a DNS zone or a record.

Format your domain name according to the following guidelines.

Contents

- [Format domain names for domain name registration](#)
- [Format domain names for DNS zones and records](#)
- [Use an asterisk \(*\) in the names of DNS zones and records](#)
- [Next steps](#)

Format domain names for domain name registration

For domain name registration, your domain name must have 1-255 characters. Valid characters for domain names include (a-z), (A-Z), (0-9), hyphens (-), and periods (.).

You can't use spaces or put a hyphen at the beginning or end of a domain name. Lightsail supports any valid generic top-level domain (TLD) name. For more information, see [Generic top-level domains](#) in the Amazon Route 53 Developer Guide.

Format domain names for DNS zones and records

For DNS zones and records, the domain name must have 1-255 characters. Valid characters for domain names include (a-z), (A-Z), (0-9), hyphens (-), and periods (.). You can't use spaces.

Lightsail stores alphabetic characters as lowercase letters (a-z), even if you specify them as uppercase letters (A-Z).

Lightsail supports DNS zones for both generic and geographic TLDs. For more examples of geographic TLDs, see [Geographic top-level domains](#) in the Amazon Route 53 Developer Guide.

Using an asterisk (*) in the names of DNS zones and records

DNS treats the asterisk (*) character as a wildcard character, depending on where the asterisk appears in the name. A wildcard DNS record is a record that answers DNS requests for any subdomain that you haven't already defined. In Lightsail, you can create DNS zones and records that include the asterisk (*) in the name with the following conditions:

DNS zones

- You can't include an asterisk (*) in the leftmost label in a domain name. For example, you can't use **subdomain*.example.com**.

- If you include the asterisk (*) in other positions, DNS treats it as an ASCII 42 character, not a wildcard. For more information about ASCII characters, see [ASCII](#) in *Wikipedia*.

DNS Records

Note the following restrictions on using an asterisk (*) as a wildcard in a DNS record name:

- As a wildcard, the asterisk must replace the leftmost label in a domain name, for example, ***.example.com** or ***.acme.example.com**. If you include an asterisk in any other position, such as **prod.*.example.com**, DNS treats it as an ASCII 42 character, not as a wildcard.
- The asterisk must replace the entire label. For example, you can't specify ***prod.example.com** or **prod.*.example.com**.
- Specific domain names take precedence. For example, if you create records for ***.example.com** and **acme.example.com**, DNS queries for **acme.example.com** respond with the values in the **acme.example.com** record.
- The asterisk applies to DNS queries for the subdomain level that includes the asterisk, and all the subdomains of that subdomain. For example, if you create a record named ***.example.com**, DNS queries for ***.example.com** will respond to the following:

zenith.example.com

acme.zenith.example.com

pinnacle.acme.zenith.example.com (if there are no records of any type for that DNS zone)

If you create a record named ***.example.com** and there's no **example.com** record, Lightsail responds to DNS queries for **example.com** with NXDOMAIN (non-existent domain).

You can configure Lightsail to return the same response to DNS queries for all subdomains at the same level and also for the domain name. For example, you can configure Lightsail to respond to DNS queries such as **acme.example.com** and **zenith.example.com** by using the **example.com** record. Perform the following steps to route traffic for subdomains to the **example.com** top-level domain:

1. Create a record for the domain, such as **example.com**.
2. Create an alias record for the subdomain, such as ***.example.com**. Specify the record that you created in the previous step as the target for the alias record.

Next steps

For more information, see the following topics:

- [Create a DNS zone to manage your domain's DNS records](#)
- [DNS](#)

Manage a Lightsail domain in Amazon Route 53

Amazon Lightsail registers domains through Amazon Route 53, a highly available and scalable DNS web service. When you register a domain using Lightsail, you can manage the domain in both Lightsail and Route 53.

Tasks such as registering a domain, and routing traffic for a domain to Lightsail resources are done in the Lightsail console. For more information, see [Domain registration in Amazon Lightsail](#).

Advanced tasks, such as transferring domains, and deleting your registration must be done in the Amazon Route 53 console.

This guide provides information for some of the advanced management tasks you can complete using the Route 53 console. For a complete overview of Route 53, see [What is Amazon Route 53?](#) in the *Amazon Route 53 Developer Guide*.

Contents

- [View the status of a domain registration](#)
- [Lock a domain to prevent unauthorized transfer to another registrar](#)
- [Restore an expired or deleted domain](#)
- [Transfer domains](#)
- [Delete a domain name registration](#)

View the status of a domain registration

Domain names have statuses that are also known as Extensible Provisioning Protocol (EPP) status codes. ICANN, the organization that maintains a central database of domain names developed the EPP status code. EPP status codes tell you the status of a variety of operations. For example, registering a domain name, renewing the registration for a domain name, and so on. All registrars

use this same set of status codes. To view the status code for your domains, see [Viewing the status of a domain registration](#) in the *Amazon Route 53 Developer Guide*.

Lock a domain to prevent unauthorized transfer to another registrar

The domain registries for all generic top-level domains (TLDs) let you lock a domain to prevent someone from transferring the domain to another registrar without your permission. For more information, see [Locking a domain to prevent unauthorized transfer to another registrar](#) in the *Amazon Route 53 Developer Guide*.

Restore an expired or deleted domain

If you don't renew a domain before the end of the late-renewal period or if you accidentally delete the domain, some registries for top-level domains (TLDs) allow you to restore the domain before it becomes available for others to register. Use the linked procedure to try to restore your domain registration. For more information, see [Restoring an expired or deleted domain](#) in the *Amazon Route 53 Developer Guide*.

Transfer domain registrations

You can transfer domain registration from another registrar to Route 53, from one AWS account to another, or from Route 53 to another registrar. For more information, see [Transfer domains](#) in the *Amazon Route 53 Developer Guide*.

Delete a domain name registration

For most top-level domains (TLDs), you can delete the registration if you no longer want it. If the registry allows you to delete the registration, perform the procedure in this topic. For more information, see [Deleting a domain name registration](#) in the *Amazon Route 53 Developer Guide*.

Provide domain information when you register or transfer a domain in Lightsail

When you use Amazon Lightsail to register a domain, you provide domain information such as the registration period (term) and domain contact information. You also configure automatic domain renewal and privacy protection.

You can also change information for a domain that is currently registered with Lightsail. Note the following:

- If you change contact information for the domain, we send an email notification to the registrant contact about the change. This email comes from *noreply@amazon.com*. For most changes, the registrant contact is not required to respond.
- For changes to contact information that also constitute a change in ownership, we send the registrant contact an additional email. ICANN, the organization that maintains a central database of domain names, requires that the registrant contact confirm receiving the email. For more information, see [First name, last name](#) and [Organization](#) later in this section.

For more information about changing contact information for an existing domain, see [Update contact information for a domain](#).

Domain information that you provide

- [Term](#)
- [Automatic domain renewal](#)
- [Registrant, administrative, and technical contacts](#)
- [Same as registrant](#)
- [Contact type](#)
- [First name, last name](#)
- [Organization](#)
- [Email](#)
- [Phone](#)
- [Address 1](#)
- [Address 2](#)
- [Country](#)
- [State](#)
- [City](#)
- [Postal/zip code](#)
- [Privacy protection](#)

Term

The registration period for the domain. The term is typically one year, although you can increase the term up to ten years while registering the domain.

Automatic domain renewal

When you register a domain with Lightsail, we configure the domain to renew automatically. The automatic renewal period is typically one year. Choose whether to have Lightsail automatically renew the domain before it expires. The registration fee is charged to your AWS account. For more information, see [Domain registration renewal](#).

Important

If you deactivate automatic domain renewal, registration for the domain will not be renewed when the expiration date passes. As a result, you might lose control of the domain name.

Registrant, administrative, and technical contacts

By default, we use the same information for all three contacts. If you want to enter different information for one or more contacts, uncheck the box next to **Same as registrant** for each contact.

Same as registrant

Specifies whether you want to use the same contact information for the registrant of the domain, the administrative contact, and the technical contact.

Contact type

Category for this contact. Note the following:

- If you choose the **Company** or **Association** option, you must enter an organization name.
- For some top-level domains (TLDs), privacy protection availability depends on the value that you choose for **Contact type**. For the privacy protection settings for your TLD, see [Domains that you can register with Amazon Route 53](#)
-

First name, last name

The first and last names of the contact. For **First name** and **Last name**, we recommend that you use the name on your official ID. For some changes to domain settings, you must provide proof of identity. In those cases, the name on your ID must match the name of the registrant contact for the domain.

If you change the email address of the registrant contact, this email is sent to both the former and new email addresses.

Organization

The organization that is associated with the contact, if any. For the registrant and administrative contacts, this is typically the organization that is registering the domain. For the technical contact, this might be the organization that manages the domain.

When the contact type is any value except **Person** and you change the **Organization** field for the registrant contact, you change the domain owner. ICANN requires that we email the registrant contact to get approval. The email comes from one of the following email addresses:

- **noreply@registrar.amazon.com**—For TLDs registered by Amazon Registrar
- **noreply@domainnameverification.net**—For TLDs registered by our registrar associate, Gandi

To determine who the registrar is for your TLD, see [Domains that you can register with Amazon Route 53](#).

If you change the email address of the registrant contact, this email is sent to both the former and new email addresses.

Email

The email address for the contact. Note the following:

If you change the email address for the registrant contact, we send notification emails to both the former and new email addresses. This email comes from **noreply@amazon.com**.

Phone

The phone number for the contact:

- If you're entering a phone number for locations in the United States or Canada, enter 1 followed by the 10-digit phone number with area code.
- If you're entering a phone number for any other location, enter the country code followed by rest of the phone number. For a list of country calling codes, see [List of country calling codes](#) on *Wikipedia*.

Address 1

The street address or P.O. box for the contact.

Address 2

Additional address information for the contact, such as apartment, suite, unit, building, floor, or mail stop.

Country

The country for the contact.

State

The state or province for the contact, if any.

City

The city for the contact.

Postal/zip code

The postal or zip code for the contact.

Privacy protection

Choose whether to conceal your contact information from WHOIS queries. If you activate privacy protection for your domain's contact information, WHOIS ("who is") queries will return contact information for the domain registrar instead of your personal information. The domain registrar is the company that manages domain name registrations.

Note

The same privacy setting applies to the administrative, registrant, and technical contacts.

If you deactivate privacy protection for your domain's contact information, you'll get more email spam at the email address that you specified.

Anyone can send a WHOIS query for a domain and get back all of the contact information for that domain. The WHOIS command is available in many operating systems, and it's also available as a web application on many websites.

Important

Although there are legitimate users for your domain contact information, the most common users are spammers, who target domain contacts with unwanted email and bogus offers. In general, we recommend leaving **Privacy protection** activated for **Contact information**.

For more information about privacy protection, see the following topics:

- [Manage privacy protection for a domain](#)
- [Domains that you can register with Amazon Route 53](#)

Manage domain registration renewal in Lightsail

When you register a domain with Amazon Lightsail, we configure the domain to renew automatically by default. The default automatic renewal period is one year, although the registries for some top-level domains (TLDs) have longer renewal periods. All generic TLDs let you extend domain registration for longer periods, typically up to ten years in one-year increments.

Note

Make sure to deactivate automatic renewal if you intend to close your AWS account. Otherwise, your domain registration will be renewed even after you have closed your account.

Contents

- [Automatic renewal](#)
- [Configure automatic renewal for a domain during domain registration](#)
- [Configure automatic renewal for a domain that is already registered](#)

Automatic renewal

The following timeline shows what happens when automatic renewal is active:

45 days before expiration

We send an email to the registrant contact to tell you that automatic renewal is active. The email also contains instructions for how to deactivate automatic renewal. Keep the registrant contact email address current so the email isn't missed.

35 or 30 days before expiration

For all domains except **.com.ar**, **.com.br**, and **.jp** domains, we renew domain registration 35 days before the expiration date. This way, we have time to resolve any issues with the renewal before the domain name expires.

The registries for **.com.ar**, **.com.br**, and **.jp** domains require that we renew the domains no more than 30 days before expiration. Gandi, our registrar associate, will send a renewal email 30 days before expiration. If automatic renewal is active, this email is sent on the same day that we renew the domain.

If automatic renewal is inactive, the following timeline shows what happens as the domain name expiration date approaches:

45 days before expiration

We send an email to inform the registrant contact that automatic renewal is currently inactive. The email also contains instructions for how to activate automatic renewal. Keep the registrant contact email address current so the email isn't missed.

35 and 7 days before expiration

If automatic renewal is inactive for the domain, ICANN, the governing body for domain registration, requires the registrar to send the registrant contact an email. The email comes from one of the following email addresses:

noreply@registrar.amazon.com—For domains with Amazon Registrar as the registrar

noreply@domainnameverification.net—For domains with our registrar associate, Gandi, as the registrar

If you activate automatic renewal less than 30 days before expiration, we renew the domain registration within 24 hours.

For more information about renewal periods, see the "Deadlines for renewing and restoring domains" section for your TLD in [Domains that you can register with Amazon Route 53](#) in the Amazon Route 53 Developer Guide.

After the expiration date

Most domains are held by the registrar for a brief time after expiration, so you might be able to renew an expired domain after the expiration date, but we strongly recommend keeping automatic renewal active if you want to keep the domain. For information about trying to renew a domain after the expiration date, see [Restore an expired or deleted domain](#) in the Amazon Route 53 Developer Guide.

If your domain expires but late renewal is allowed for the domain, you can renew the domain for the standard renewal price. To determine whether a domain is still within the late-renewal period, perform the procedure in [Extend the registration period for a domain](#) in the Amazon Route 53 Developer Guide. If the domain is still listed, it's within the late-renewal period.

Configure automatic renewal for a domain during domain registration

When you register a new domain name with Lightsail, we configure the domain to renew automatically. You can choose to deactivate automatic domain renewal during the domain registration procedure.

1. Sign in to the [Lightsail console](#).
2. Choose the **Domains & DNS** tab.

3. Choose the **Register domain** button.
4. Specify the domain name that you want to register with Lightsail, then choose **Check availability**.
5. If the domain name is available, you will see the domain registration page. In the **Automatic domain renewal** section, turn the toggle switch on or off to activate or deactivate automatic domain renewal.

Configure automatic renewal for a domain that is already registered

When you want to change whether Lightsail automatically renews registration for a domain shortly before the expiration date, or if you want to view the current setting for automatic renewal, perform the following procedure.

1. Sign in to the [Lightsail console](#).
2. Choose the **Domains & DNS** tab.
3. Choose the domain that you want to view or update.
4. Choose the **Contact info** tab
5. In the **Automatic domain renewal** section, turn the toggle switch on or off to activate or deactivate automatic renewal for the domain's registration period.

Manage privacy protection for domain contacts in Lightsail

When you register a domain with Lightsail, we activate privacy protection by default for all the domain contacts. This typically hides most of your contact information from WHOIS ("Who is") queries and reduces the amount of spam that you receive. Your contact information is replaced with either the contact information for the registrar or with the phrase "REDACTED FOR PRIVACY." There are no charges for using privacy protection.

If you choose to deactivate privacy protection, anyone can send a WHOIS query for the domain and, for most top-level domains (TLDs), they might be able to get all the contact information that you provided when you registered the domain. This information includes name, address, phone number, and email address. The WHOIS command is widely available. It's included in many operating systems, and it's also available as a web application on many websites.

To manage privacy protection for a domain that you registered by using Lightsail, perform the following procedure.

Contents

- [Complete the prerequisites](#)
- [Manage privacy protection for your domain](#)

Complete the prerequisites

Register a domain with Lightsail. For more information, see [Register a new domain](#).

Manage privacy protection for your domain

1. Sign in to the [Lightsail console](#).
2. Choose the **Domains & DNS** tab.
3. Choose the name of the domain that you want to change the privacy protection for.
4. Choose **Contact info**.
5. You can manage privacy protection for your contact information by turning the **Privacy protection** toggle switch on or off.

Update contact information for a domain in Lightsail

When you register a domain with Amazon Lightsail, you specify contact information for your domain. Following are three types of contact information:

- **Registrant:** Owner of the domain
- **Administrator:** Person responsible for administering your domain
- **Technical:** Person responsible for making technical changes to your domain

Your domain's contact information is used to verify ownership of your domain and to keep you updated about any information related to your domain name.

Topics

- [Who is the owner of a domain?](#)
- [Update contact information for a domain](#)

Who is the owner of a domain?

When the contact type is **Person** and you change the **First Name** or **Last Name** fields for the registrant contact, you change the owner of the domain.

When the contact type is any value except **Person** and you change **Organization**, you change the owner of the domain.

The following actions happen when you change the contact information for a domain that is currently registered with Lightsail:

- If you change contact information for the domain, we send an email notification to the registrant contact about the change. This email comes from **noreply@amazon.com**. For most changes, the registrant contact is not required to respond.
- For changes to contact information that also constitute a change in ownership, we send the registrant contact an additional email. ICANN, the organization that maintains a central database of domain names, requires that the registrant contact confirm receiving the email.

Update contact information for a domain

To update contact information for a domain, perform the following procedure.

1. Sign in to the [Lightsail console](#).
2. Choose the **Domains & DNS** tab.
3. Choose the name of the domain that you want to update.
4. Choose the **Contact info** tab. Then, choose **Edit contact**.
5. Update the applicable values. For more information, see [Values that you specify when you register or transfer a domain](#) in the Amazon Route 53 Developer Guide.
6. Choose **Save**.

Databases in Amazon Lightsail

You can create a MySQL or PostgreSQL managed database in Amazon Lightsail with a few steps. Lightsail makes database administration more efficient by managing your common maintenance and security tasks. Using the Lightsail console, you can:

- Back up your database to a snapshot.
- Create a new larger database from a snapshot.
- Troubleshoot common issues with browser-based logs and metrics.
- Recover data by using point-in-time backup and restore operations.

You can build your application on a Lightsail instance and connect it to a Lightsail managed database. You can also create a standalone database, and connect analytics or querying tools for your company. Choose from standard or high availability database plans that include your pre-configured database, SSD-based storage, and data transfer allocation for a fixed, monthly price. You can also manage Lightsail databases using the AWS Command Line Interface (AWS CLI), API, or SDK.

Choose a Lightsail database

Amazon Lightsail provides the latest major versions of the MySQL and PostgreSQL databases. This guide helps you decide which database is right for your project.

Lightsail also offers a Windows Server 2022 instance with SQL Server. For more information, see [Choose an Amazon Lightsail instance image](#).

Compare managed databases in Lightsail

MySQL

MySQL 5.7, and 8.0 are available in Lightsail. MySQL is the most widely adopted open source relational database. It serves as the primary relational data store for many popular websites, applications, and commercial products. MySQL is a reliable, stable, and secure SQL-based database management system, with more than 20 years of community-backed development and support. The MySQL database is suitable for a wide variety of use cases, including mission-critical apps and dynamic websites. It also functions as an embedded database for software, hardware, and appliances.

⚠ Important

Starting June 30, 2024, Lightsail will no longer support MySQL 5.7, and you will not be able to create new databases with this blueprint. To learn how you can upgrade major versions of your database instance, see [Upgrade the major version of a Lightsail database](#).

For more information, see the following MySQL documentation:

- [MySQL 5.7 documentation](#)
- [MySQL 8.0 documentation](#)

PostgreSQL

PostgreSQL 11, 12, 13, 14, 15, and 16 are available in Lightsail. PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

There is a wealth of information to be found describing how to install and use PostgreSQL through the [official documentation](#). The [PostgreSQL community](#) provides many helpful places to become familiar with the technology, discover how it works, and find career opportunities.

⚠ Important

Starting June 30, 2024, Lightsail will no longer support PostgreSQL 11, and you will not be able to create new databases with this blueprint. To learn how you can upgrade major versions of your database instance, see [Upgrade the major version of a Lightsail database](#).

For more information, see the following PostgreSQL documentation:

- [PostgreSQL 11 documentation](#)
- [PostgreSQL 12 documentation](#)
- [PostgreSQL 13 documentation](#)
- [PostgreSQL 14 documentation](#)
- [PostgreSQL 15 documentation](#)
- [PostgreSQL 16 documentation](#)

Optimize data import

Several database plans are available in Lightsail, each with specific memory, vCPU, storage, and data transfer allowance specifications. Because each database plan has these specifications, it is important that you choose an appropriately-sized database plan for the amount of data that you want to import into your new Lightsail database. Your data import may be slowed if you choose a plan that is under your size requirements. Use the following guidelines to select the appropriate database plan for your data import requirement:

- **Micro \$15 USD/month database plan** — Data import may be slowed if you transfer more than 10 GB of data.
- **Small \$30 USD/month database plan** — Data import may be slowed if you transfer more than 20 GB of data.
- **Medium \$60 USD/month database plan** — Data import may be slowed if you transfer more than 85 GB of data.
- **Large \$115 USD/month database plan** — Data import may be slowed if you transfer more than 156 GB of data.

Note

For more information about importing data into your database, see [Import data into your MySQL database](#) or [Import data into your PostgreSQL database](#).

High availability databases in Lightsail

A Lightsail high availability managed database provides failover support with a primary database in one Availability Zone, and a secondary standby database in another. We recommend high availability databases for production workloads that experience heavy use and require data redundancy. For development and test purposes, you can use a standard database that isn't high availability.

To create a high availability database, select one of the high availability database plans available in Lightsail when creating your managed database. For more information, see [Create a database](#). You can also change your standard database to a high availability database. Create a snapshot of your standard database, create a new database from the snapshot, and choose a high availability plan. For more information, see [Create a database from a snapshot](#).

Create a Lightsail database

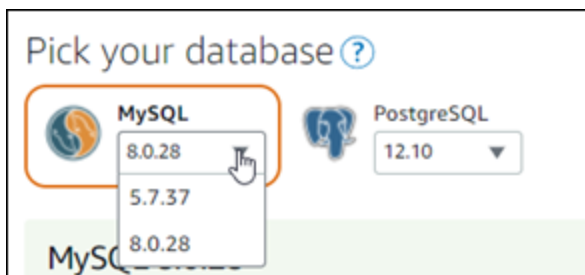
Create a managed database in Amazon Lightsail in minutes. You can choose between the latest major versions of MySQL or PostgreSQL, and configure your database with a standard or high availability plan.

Note

For more information about managed databases in Lightsail, see [Choose a database](#).

To create a database

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose **Create database**.
4. Choose the AWS Region and Availability Zone for your database.
 1. Choose **Change AWS Region and Availability Zone**, then choose a Region.
 2. Choose **Change your Availability Zone**, then choose an Availability Zone.
5. Choose your database type. Under one of the database engine options available, choose the drop-down menu, and then choose one of the latest major database versions supported by Lightsail.



6. If necessary, choose one of these options:
 - **Specify login credentials** — Specify your own database user name and password. Otherwise, Lightsail specifies the user name, and creates a strong password for you.
 - To specify your own user name, choose **Specify login credentials**, and enter your user name into the text box. The following constraints apply according to the database engine you select:

MySQL

- Required for MySQL.
- Must be 1 to 16 letters or numbers.
- First character must be a letter.
- Can't be a reserved word for the chosen database engine. For more information about reserved words in MySQL, see the Keywords and Reserved Words articles for [MySQL 5.6](#), [MySQL 5.7](#), or [MySQL 8.0](#).

PostgreSQL

- Required for PostgreSQL.
- Must be 1 to 63 letters or numbers.
- First character must be a letter.
- Can't be a reserved word for the chosen database engine. For more information about reserved words in PostgreSQL, see the SQL Key Words articles for [PostgreSQL 9.6](#), [PostgreSQL 10](#), [PostgreSQL 11](#), or [PostgreSQL 12](#).
- To specify your own password, clear the **Create a strong password for me** check box, and enter your password into the text box. The password can include any printable ASCII character except "/", "", or "@". For MySQL databases, the password can contain from 8 to 41 characters. For PostgreSQL databases, the password can contain from 8 to 128 characters.
- **Specify the master database name** — Specify your own primary database name, or Lightsail specifies the name for you. To specify your own primary database name, choose **Specify the master database name**, and enter a name into the text box. The following constraints apply according to the database engine you select:

MySQL

- Must contain 1 to 64 letters or numbers.
- Must begin with a letter. Subsequent characters can be letters, underscores, or digits (0-9).
- Can't be a reserved word for the chosen database engine. For more information about reserved words in MySQL, see the Keywords and Reserved Words articles for [MySQL 5.6](#), [MySQL 5.7](#), or [MySQL 8.0](#).

PostgreSQL

- Must contain 1 to 63 letters, numbers, or underscores.

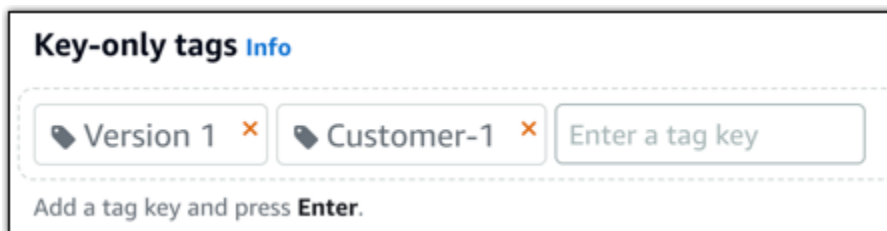
- Must begin with a letter. Subsequent characters can be letters, underscores, or digits (0-9).
 - Can't be a reserved word for the chosen database engine. For more information about reserved words in PostgreSQL, see the SQL Key Words articles for [PostgreSQL 9.6](#), [PostgreSQL 10](#), [PostgreSQL 11](#), or [PostgreSQL 12](#).
7. Choose a high availability or a standard database plan.

A database created with a high availability plan has a primary database and a secondary standby database in another Availability Zone for failover support. For more information, see [High availability databases](#). Differently priced database bundle options are available, each with different levels of memory, processing, storage space, and transfer rates.

8. Enter a name for your database.

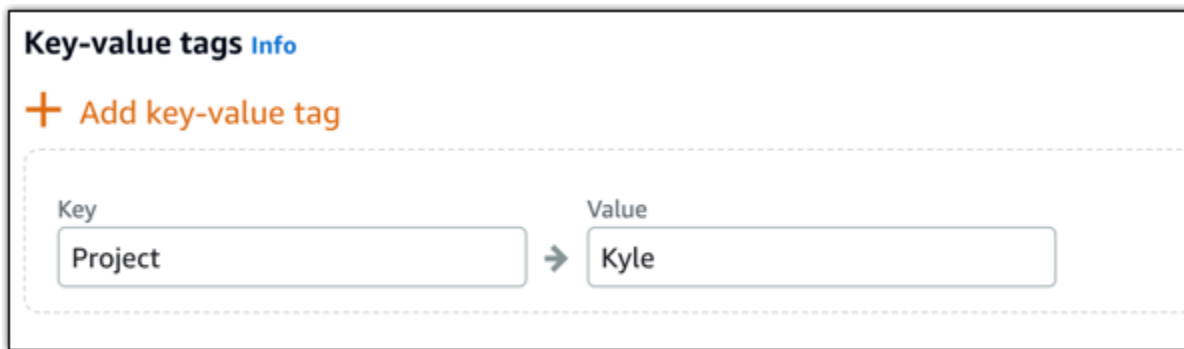
Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
9. Choose one of the following options to add tags to your database:
- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key: Project → Value: Kyle

Note

For more information about key-only and key-value tags, see [Tags](#).

10. Choose Create database.

Within minutes, your Lightsail database is ready. You can begin configuring it for data import, or connect to it by using a database client.

Next steps

Here are a few guides to help you manage your new database in Lightsail after it's up and running:


- [Configure the data import mode for your database](#)
- [Configure the public mode for your database in Amazon Lightsail](#)
- [Manage your database password](#)
- [Connect to your MySQL database](#)
- [Connect to your PostgreSQL database](#)
- [Import data into your MySQL database](#)
- [Import data into your PostgreSQL database](#)
- [Create a snapshot of your database](#)

Connect to your Lightsail MySQL database

After your MySQL managed database is created in Amazon Lightsail, you can use any standard MySQL client application or utility to connect to it. You must get the database endpoint, port, user

name, and password from your database management page in the Lightsail console. Specify those values when configuring the database connection in your client or web application.

This guide shows you how to obtain the required connection information, and how to configure MySQL Workbench to connect to your managed database.

 **Note**

For more information about connecting to a PostgreSQL database, see [Connect to your PostgreSQL database](#).

Step 1: Get your MySQL database connection details

Get your database endpoint and port information from the Lightsail console. You use these later when configuring your client to connect to your database.

To get your database connection details

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database that you want to connect to.
4. On the **Connect** tab, under the **Endpoint and port** section, note the endpoint and port information.

We recommend copying the endpoint to your clipboard to avoid entering it incorrectly. To do that, highlight the endpoint and press **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using macOS, to copy it to your clipboard. Then, press **Ctrl+V** or **Cmd+V** as appropriate to paste it.



5. On the **Connect** tab, under the **User name and passwords** section, make note of the user name, then choose **Show** under the **Password** section to view the current database password.

Because managed passwords are complex, we also recommend copying and pasting it to avoid entering it incorrectly. Highlight the managed password and press **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using macOS, to copy it to your clipboard. Then, press **Ctrl+V** or **Cmd+V** as appropriate to paste it.

Step 2: Configure the public availability of your MySQL database

You must enable public mode for your database to connect to it externally, or from a Lightsail instance in a different AWS Region than your database. With public mode enabled, anyone with the database user name and password can connect to your database. To configure the public availability of your database, follow the steps in the [Configure the public mode for your database](#) guide.

Note

Skip to step 3 if you plan to connect to your database from one of your Lightsail instances that is in the same Region as your database.

Step 3: Configure your database client to connect to your MySQL database

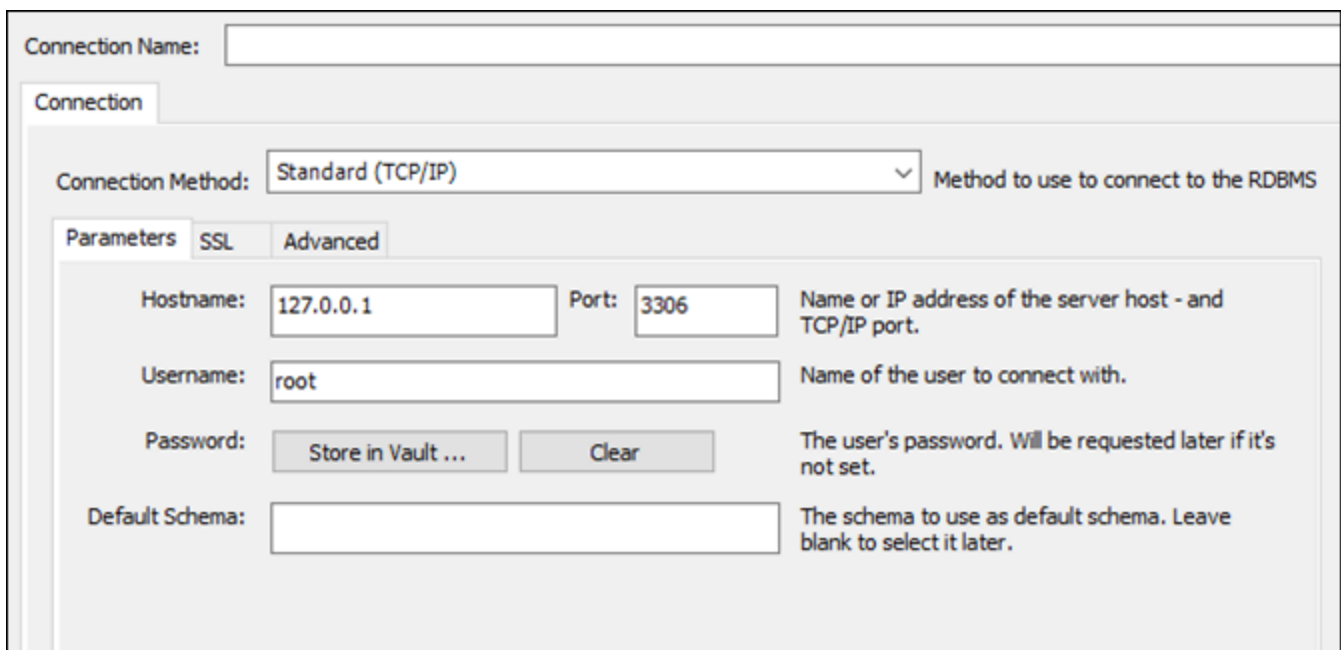
To connect to your MySQL database, configure your database client to use the endpoint and port that you obtained earlier. The following steps show you how to configure MySQL Workbench, but these steps may be similar for other clients.

Note

For more information about using MySQL Workbench, see the [MySQL Workbench Manual](#).

To configure MySQL Workbench to connect to your database

1. Open **MySQL Workbench**.
2. Choose the **Database** menu, then choose **Manage connections**.
3. Enter the following information into the form that displays:



Connection Name:

Connection

Connection Method: Method to use to connect to the RDBMS

Parameters

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

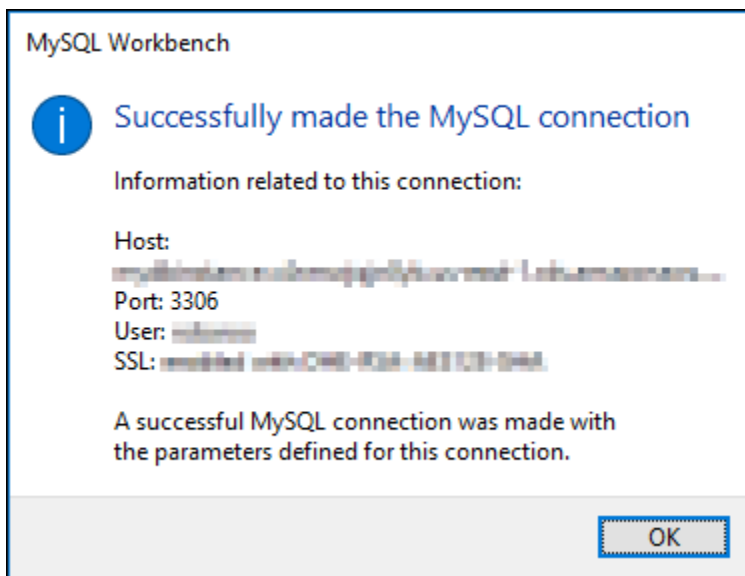
Password: The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

- **Connection Name** — We recommend using a name for the connection that is similar to your database. This helps you identify it in the future.
- **Connection Method** — Choose **Standard (TCP/IP)**.
- **Port** — Enter the port for your database that you obtained earlier. The default port for MySQL is 3306.

- **Hostname** — Enter the database endpoint that you obtained earlier. If you copied the database endpoint from the Lightsail console, and it's still in your clipboard, press **Ctrl+V** if you're using Windows, or **Cmd+V** if you're using macOS, to paste it.
 - **Username** — Enter the database user name that you obtained earlier.
 - **Password** — Choose **Store in vault**. In the window that appears, enter your database password that you obtained earlier. If you copied your password from the Lightsail console, and it's still in your clipboard, press **Ctrl+V** if you're using Windows, or **Cmd+V** if you're using macOS, to paste it. Choose **OK** to save your password.
 - **Default Schema** — Keep this text box blank.
4. Choose **Test connection** to determine if the client can establish a connection with your database.

If the connection is successful, a prompt similar to the following example displays. After you read the information, choose **OK** to close it.

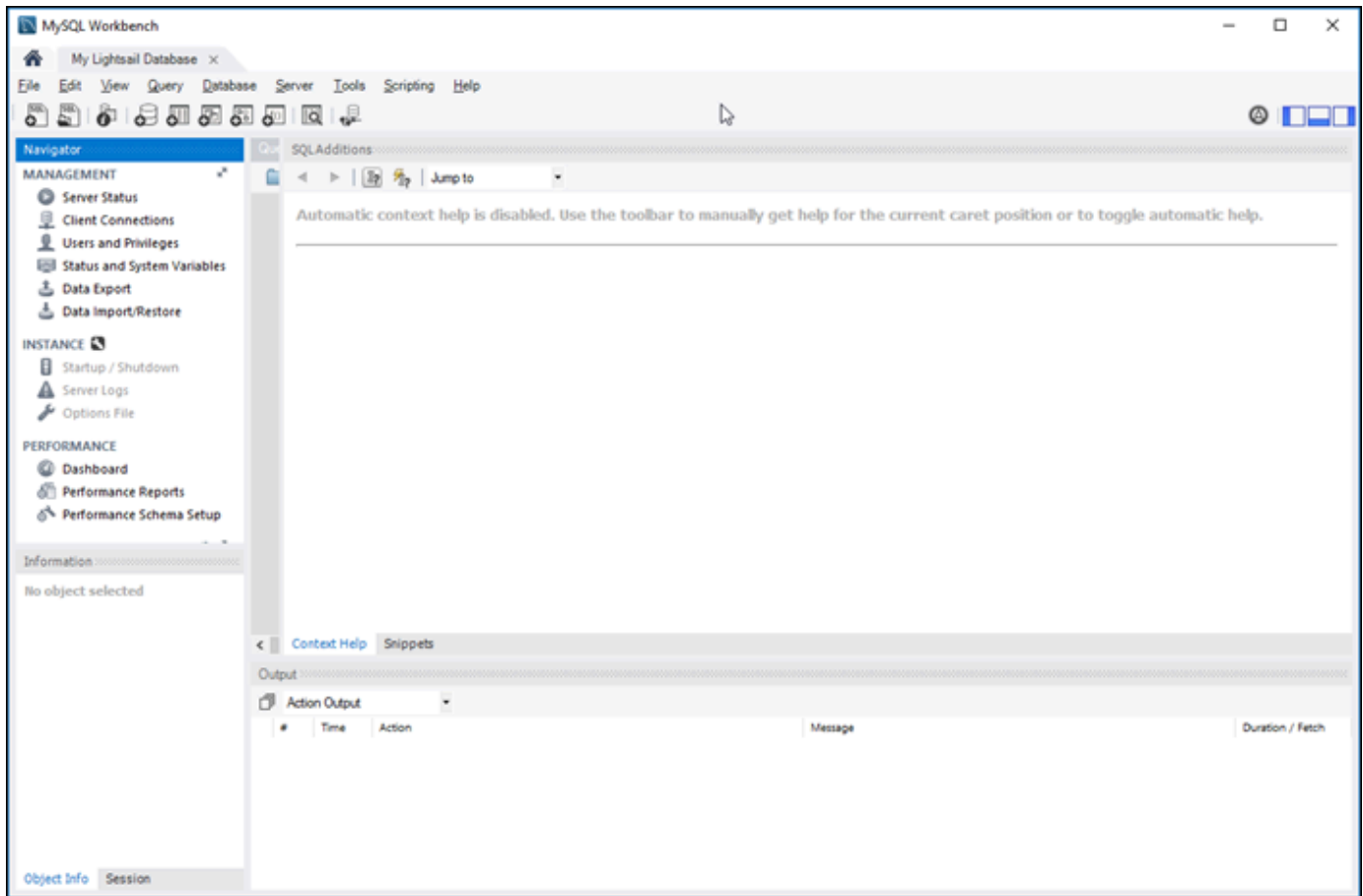


5. Choose **New** to save the new connection details, then choose **Close** to close the connections management window.

Your new database connection appears on the home page of the MySQL Workbench application, under the MySQL Connections section.

6. To connect to your database, choose your new database connection.

If the connection is successful, a window similar to the following example displays.



Next steps

Here is a guide to help you import data into your database in Lightsail:

- [Import data into your MySQL database](#)

Connect to your Lightsail MySQL database using SSL

Amazon Lightsail creates an SSL certificate, and installs it on your MySQL managed database when it's provisioned. The certificate is signed by a certificate authority (CA), and it includes the database endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.

An SSL certificate created by Lightsail is the trusted root entity and should work in most cases but might fail if your application does not accept certificate chains. If your application does not accept certificate chains, you might need to use an intermediate certificate to connect to your AWS Region.

For more information about the CA certificates for your managed database, supported AWS Regions, and how you can download intermediate certificates for your applications, see [Download an SSL certificate for your managed database](#).

Supported connections

MySQL uses yaSSL for secure connections in the following versions:

- MySQL version 5.7.19 and earlier 5.7 versions
- MySQL version 5.6.37 and earlier 5.6 versions
- MySQL version 5.5.57 and earlier 5.5 versions

MySQL uses OpenSSL for secure connections in the following versions:

- MySQL version 8.0
- MySQL version 5.7.21 and later 5.7 versions
- MySQL version 5.6.39 and later 5.6 versions
- MySQL version 5.5.59 and later 5.5 versions

MySQL managed databases support Transport Layer Security (TLS) versions 1.0, 1.1, and 1.2. The following list shows the TLS support for MySQL versions:

- MySQL 8.0—TLS1.0, TLS 1.1, and TLS 1.2
- MySQL 5.7—TLS1.0, and TLS 1.1. TLS 1.2 is supported only for MySQL 5.7.21 and later.
- MySQL 5.6—TLS1.0
- MySQL 5.5—TLS1.0

Prerequisites

- Install MySQL server on the computer you will use to connect to your database. For more information, see [MySQL Community Server download](#) in the MySQL website.
- Download the appropriate certificate for your database. For information, see [Download an SSL certificate for your managed database](#).

Connect to your MySQL database using SSL

Complete the following steps to connect to your MySQL database using SSL.

1. Open a Terminal or Command Prompt window.
2. Enter one of the following commands depending on the version of your MySQL database:
 - Enter the following command to connect to a database that is MySQL 5.7 or later.

```
mysql -h DatabaseEndpoint --ssl-ca=/path/to/certificate/rds-combined-ca-bundle.pem --ssl-mode=VERIFY_IDENTITY -u UserName -p
```

In the command, replace:

- *DatabaseEndpoint* with the endpoint of your database.
- */path/to/certificate/rds-combined-ca-bundle.pem* with the local path where you downloaded and saved the certificate for your database.
- *UserName* with the user name of your database.

Example:

```
mysql -h ls-1c51a7c70a4fb55e542829a4e4e0d735ba42.czowadgeezqi.us-west-2.rds.amazonaws.com --ssl-ca=/home/ec2-user/rds-combined-ca-bundle.pem --ssl-mode=VERIFY_IDENTITY -u dbmasteruser -p
```

- Enter the following command to connect to a database that is MySQL 6.7 or earlier.

```
mysql -h DatabaseEndpoint --ssl-ca=/path/to/certificate/rds-combined-ca-bundle.pem --ssl-verify-server-cert -u UserName -p
```

In the command, replace:

- *DatabaseEndpoint* with the endpoint of your database.
- */path/to/certificate/rds-combined-ca-bundle.pem* with the local path where you downloaded and saved the certificate for your database.
- *UserName* with the user name of your database.

Example:


```
mysql -h ls-1c51a7c70a4fb55e542829a4e4e0d735ba42.czowadgeezqi.us-west-2.rds.amazonaws.com --ssl-ca=/home/ec2-user/rds-combined-ca-bundle.pem --ssl-verify-server-cert -u dbmasteruser -p
```

3. Type the password for the database user you specified in the previous command when prompted, and press **Enter**.

You should see a result similar to the following example:

```
[ec2-user@ip-172-26-5-44 ~]$ mysql -h ls-1c51a7beedc70a4fb55e542829a4e4e0d735ba42.czowadgeezqi.us-west-2.rds.amazonaws.com --ssl-ca=/home/ec2-user/rds-ca-2015-root.pem --ssl-verify-server-cert -u dbmasteruser -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2727
Server version: 8.0.16 Source distribution

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

4. Type **status**, and press **Enter** to view the status of your connection.

Your connection is encrypted if you see a value of “Cipher in use is” next to SSL.

```
mysql> status
-----
mysql Ver 14.14 Distrib 5.5.62, for Linux (x86_64) using readline 5.1

Connection id:          2727
Current database:
Current user:           dbmasteruser@ip-172-26-5-44
SSL:                    Cipher in use is DHE-RSA-AES256-SHA
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         8.0.16 Source distribution
Protocol version:       10
Connection:             ls-1c51a7beedc70a4fb55e542829a4e4e0d735ba42.czowadgeezqi.us-west-2.rds.amazonaws.com via TCP/IP
Server characterset:    utf8mb4
Db characterset:        utf8mb4
Client characterset:    utf8
Conn. characterset:     utf8
TCP port:               3306
Uptime:                 9 days 16 hours 24 min 33 sec

Threads: 3  Questions: 557480  Slow queries: 0  Opens: 242  Flush tables: 3  Open tables: 146  Queries per second avg: 0.666
-----
```

Connect to your Lightsail PostgreSQL database

After your PostgreSQL managed database is created in Amazon Lightsail, you can use any standard PostgreSQL client application or utility to connect to it. You must get the database endpoint, port,

user name, and password from your database management page in the Lightsail console. Specify those values when configuring the database connection in your client or web application.

This guide shows you how to obtain the required connection information, and how to configure the pgAdmin client to connect to your managed database.

Note

For more information about connecting to a MySQL database, see [Connect to your MySQL database](#).

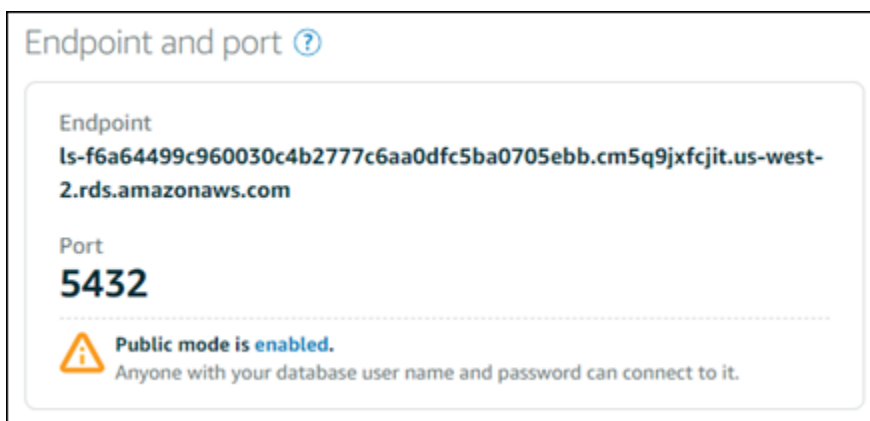
Step 1: Get your PostgreSQL database connection details

Get your database endpoint and port information from the Lightsail console. You use these later when configuring your client to connect to your database.

To get your database connection details

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database that you want to connect to.
4. On the **Connect** tab, under the **Endpoint and port** section, note the endpoint and port information.

We recommend copying the endpoint to your clipboard to avoid entering it incorrectly. To do that, highlight the endpoint and press **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using macOS, to copy it to your clipboard. Then, press **Ctrl+V** or **Cmd+V** as appropriate to paste it.



5. On the **Connect** tab, under the **User name and passwords** section, make note of the user name, then choose **Show** under the **Password** section to view the current database password.

Because managed passwords are complex, we also recommend copying and pasting it to avoid entering it incorrectly. Highlight the managed password and press **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using macOS, to copy it to your clipboard. Then, press **Ctrl+V** or **Cmd+V** as appropriate to paste it.

Step 2: Configure the public availability of your PostgreSQL database

You must enable public mode for your database to connect to it externally, or from a Lightsail instance in a different Region than your database. With public mode enabled, anyone with the database user name and password can connect to your database. To configure the public availability of your database, follow the steps in the [Configure the public mode for your database](#) guide.

Note

Skip to step 3 if you plan to connect to your database from one of your Lightsail instances that is in the same Region as your database.

Step 3: Configure your database client to connect to your PostgreSQL database

To connect to your PostgreSQL database, configure your database client to use the endpoint and port that you obtained earlier. The following steps show you how to configure pgAdmin, but these steps may be similar for other clients.

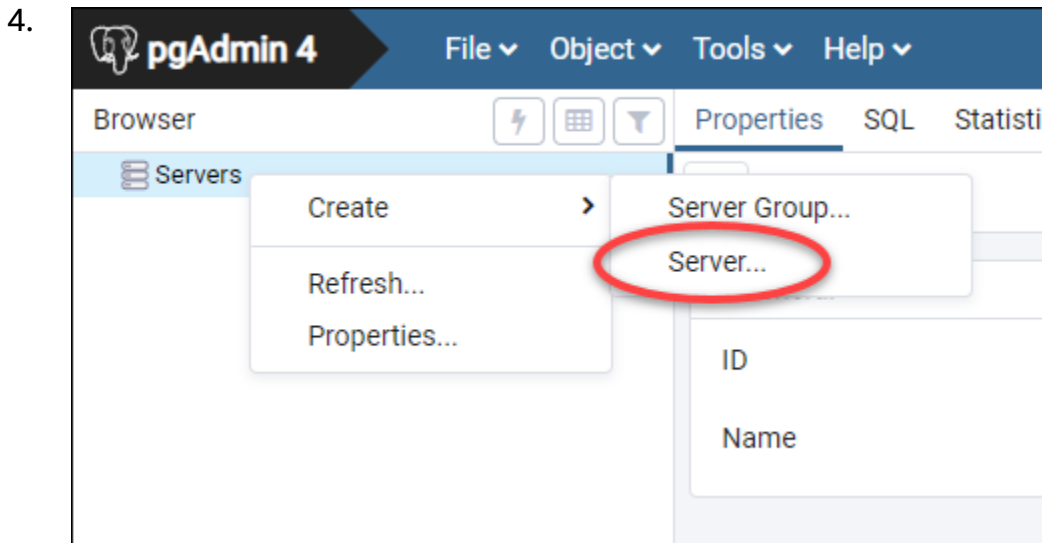
Note

For more information about using pgAdmin, see the [pgAdmin Documentation](#).

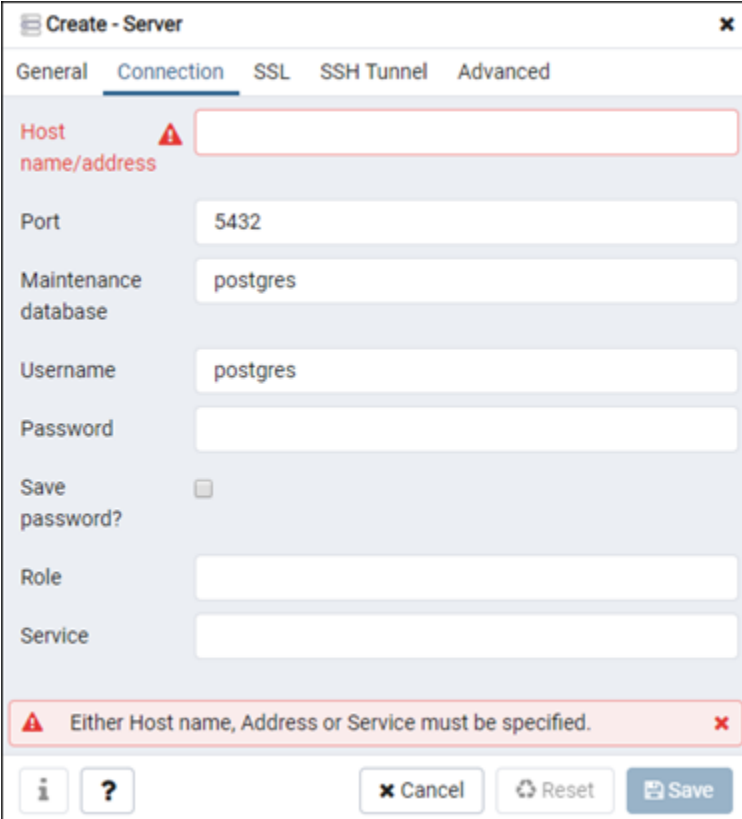
To configure pgAdmin to connect to your database

1. Open **pgAdmin**.

2. Right-click **Servers** from the left navigation menu.
3. Choose **Create**, then choose **Server**.



5. In the **Create - Server** form, enter a name for the server. We recommend using a name for the connection that is similar to your database. This helps you identify it in the future.
6. Choose the **Connection** tab, then enter the following information into the form that displays:

A screenshot of the 'Create - Server' dialog box in pgAdmin 4. The dialog has a title bar 'Create - Server' and a close button. It features several tabs: 'General', 'Connection', 'SSL', 'SSH Tunnel', and 'Advanced'. The 'Connection' tab is selected. The form contains the following fields:

- Host name/address**: A text input field with a red warning icon to its left.
- Port**: A text input field containing the value '5432'.
- Maintenance database**: A text input field containing the value 'postgres'.
- Username**: A text input field containing the value 'postgres'.
- Password**: A text input field.
- Save password?**: A checkbox that is currently unchecked.
- Role**: A text input field.
- Service**: A text input field.

At the bottom of the form, there is a red error message: 'Either Host name, Address or Service must be specified.' Below the error message are three buttons: 'Cancel', 'Reset', and 'Save'.

- **Host name/address** — Enter the database endpoint that you obtained earlier. If you copied the database endpoint from the Lightsail console, and it's still in your clipboard, press **Ctrl+V** if you're using Windows, or **Cmd+V** if you're using macOS, to paste it.
- **Port** — Enter the port for your database that you obtained earlier. The default port for PostgreSQL is 5432.
- **Maintenance database** — Specify the name of the initial database to which the client will connect. This is the primary database name that you specified when you created your PostgreSQL database in Lightsail.

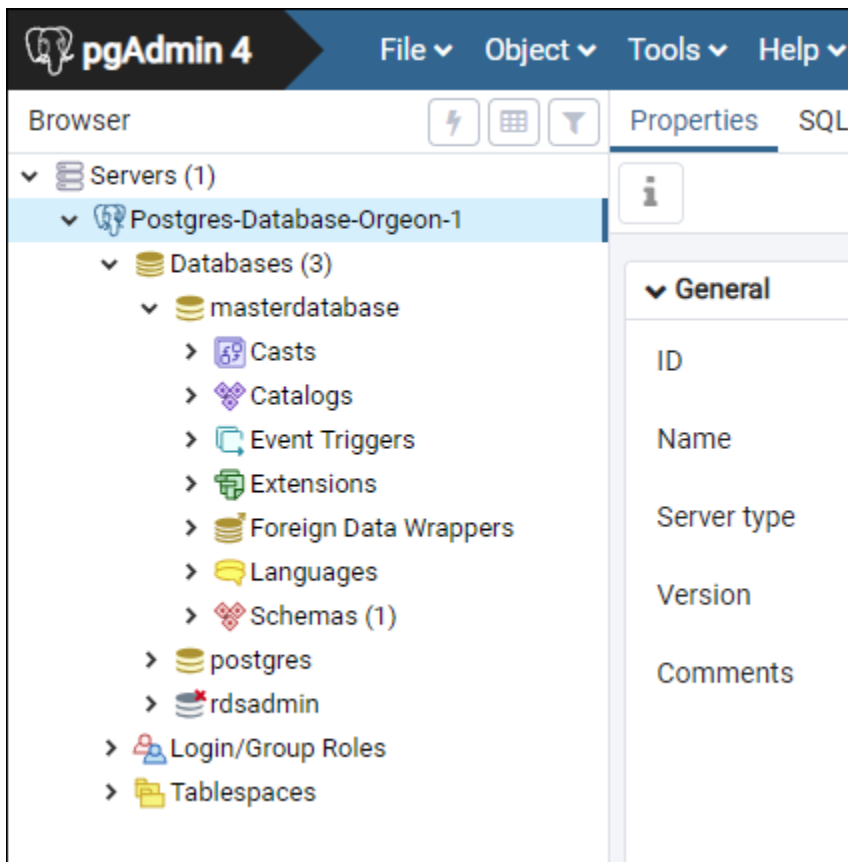
Enter `postgres` if you can't remember the name of your primary database. Every PostgreSQL managed database has a `postgres` database that you can connect to, after which you'll be able to access all other databases on the PostgreSQL managed database.

- **Username** — Enter the database user name that you obtained earlier.
 - **Password** — Enter your database password that you obtained earlier. If you copied your password from the Lightsail console, and it's still in your clipboard, press **Ctrl+V** if you're using Windows, or **Cmd+V** if you're using macOS, to paste it. Choose **Save password** to save your password.
 - **Role and Service** — Leave these fields blank.
7. Choose **Save** to save the new server details.

Your new database connection appears on the left navigation menu of the pgAdmin application, under the Servers section.

8. To connect to your database, double-click your new database connection.

If the connection is successful, you will see a list of available resources for that database.



Next steps

Here is a guide to help you import data into your database in Lightsail:

- [Import data into your PostgreSQL database](#)

Connect to your Lightsail PostgreSQL database using SSL

Amazon Lightsail creates an SSL certificate, and installs it on your PostgreSQL (Postgres) managed database when it's provisioned. The certificate is signed by a certificate authority (CA), and it includes the database endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.

An SSL certificate created by Lightsail is the trusted root entity and should work in most cases but might fail if your application does not accept certificate chains. If your application does not accept certificate chains, you might need to use an intermediate certificate to connect to your AWS Region.

For more information about the CA certificates for your managed database, supported AWS Regions, and how you can download intermediate certificates for your applications, see [Download an SSL certificate for your managed database](#).

Prerequisites

- Install PostgreSQL server on the computer you will use to connect to your database. For more information, see [PostgreSQL Downloads](#) in the Postgres website
- Download the appropriate certificate for your database. For information, see [Download an SSL certificate for your managed database](#).

Connect to your Postgres database using SSL

Complete the following steps to connect to your Postgres database using SSL.

1. Open a Terminal or Command Prompt window.
2. Enter the following command to connect to a PostgreSQL database.

```
psql -h DatabaseEndpoint -p 5432 "dbname=DatabaseName user=UserName sslrootcert=  
/path/to/certificate/rds-combined-ca-bundle.pem sslmode=verify-full"
```

In the command, replace:

- *DatabaseEndpoint* with the endpoint of your database.
- *DatabaseName* with the name of the database you want to connect to.
- *UserName* with the user name of your database.
- */path/to/certificate/rds-combined-ca-bundle.pem* with the local path where you downloaded and saved the certificate for your database.

Example:

```
psql -h ls-8e81e07f8b821917b11e1c6a0e26cb73c203.czowadgeezqi.us-  
west-2.rds.amazonaws.com -p 5432 "dbname=dbmaster user=dbmasteruser sslrootcert=  
/home/ec2-user/rds-combined-ca-bundle.pem sslmode=verify-full"
```

3. Type the password for the database user you specified in the previous command when prompted, and press **Enter**.

You should see a result similar to the following example. Your connection is encrypted if you see a value of "SSL connection."

```
[ec2-user@ip-172-31-26-115 ~]$ psql -h ls-8e81e04e807f8b821917b11e1c6a0e26cb73c203.czowadgeezqi.us-west-2.rds.amazonaws.com -p 5432 "dbname=dbmaster user=dbmaster sslrootcert=/home/ec2-user/rds-combined-ca-bundle.pem sslmode=verify-full"
Password:
psql (10.4, server 11.5)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
type "help" for help.

dbmaster=> |
```

Delete your Lightsail database

Delete your managed database in Amazon Lightsail if you no longer need it. You stop incurring charges for the database as soon as it's deleted.

Note

You can't recover a deleted database. You can create a final snapshot of your database as part of the steps covered in this guide, or you can create a snapshot separately from the deletion process. For more information, see [Create a snapshot of your database](#).

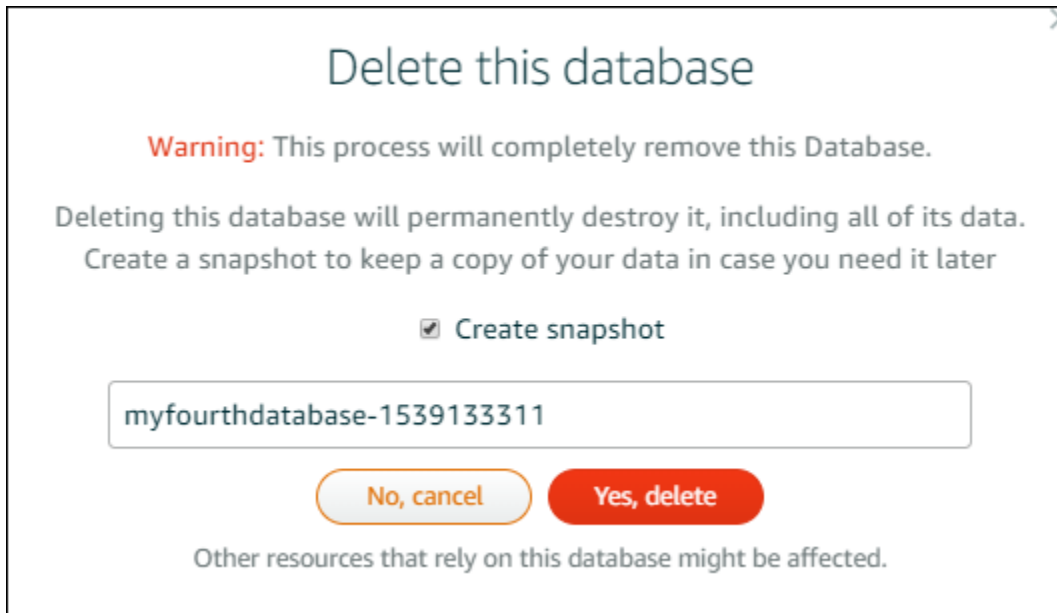
To delete your database

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database that you want to delete.
4. Choose the **Delete** tab.
5. Add a check mark next to **Create snapshot before deletion** to create a final snapshot before deleting the database. Then enter a name for your snapshot.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

6. Choose **Delete database**.
7. Choose **Yes, delete** to confirm the deletion.



If you opted to create a snapshot before deleting, you can view it on the **Snapshots** tab of the Lightsail home page.

Configure the data import mode for your Lightsail database

Regular database backup operations can cause substantial delays, or slowdowns, when importing large amounts of data all at once. Enable the data import mode for your Amazon Lightsail managed database to suspend these operations while you import large amounts of data.

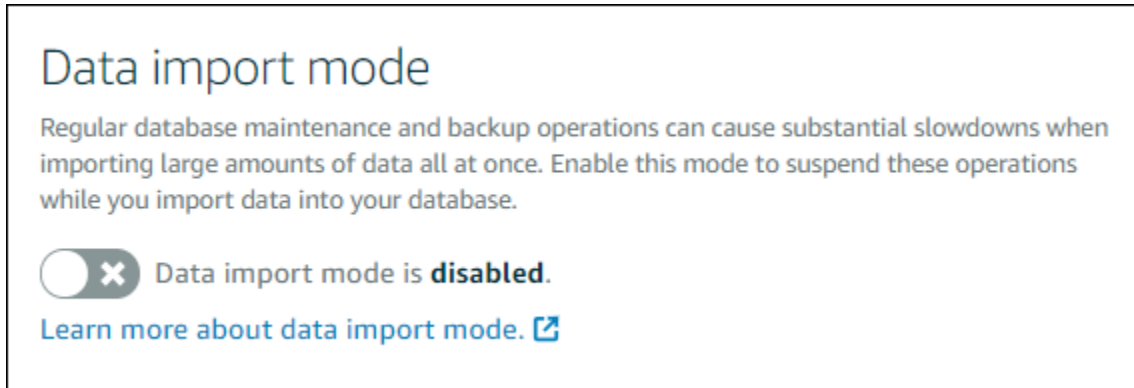
⚠ Important

All emergency restore backups are deleted when data import mode is enabled. Create a snapshot of your database if you would like to have a backup before data import mode is enabled. For more information, see [Create a snapshot of your database](#).

To configure the data import mode for your database

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.

3. Choose the name of the database for which you want to configure data import mode.
4. On the **Connect** tab, under the **Data import mode** section, use the toggle to turn on the data import mode. Likewise, after the import is complete, use the toggle to turn it off.



Now that the data import mode is enabled, database backup operations are suspended. We recommend that you enable data import mode temporarily. Use it only when it's necessary for you to import large amounts of data into your database. Disable data import mode as soon as you're done to restore backup operations.

Note

Your import may be slowed depending on the amount of data that you're importing. For more information, see [Optimizing data import](#).

Import data into your MySQL database in Lightsail

You can import a SQL file (.SQL) into your MySQL managed database in Amazon Lightsail using MySQL Workbench.

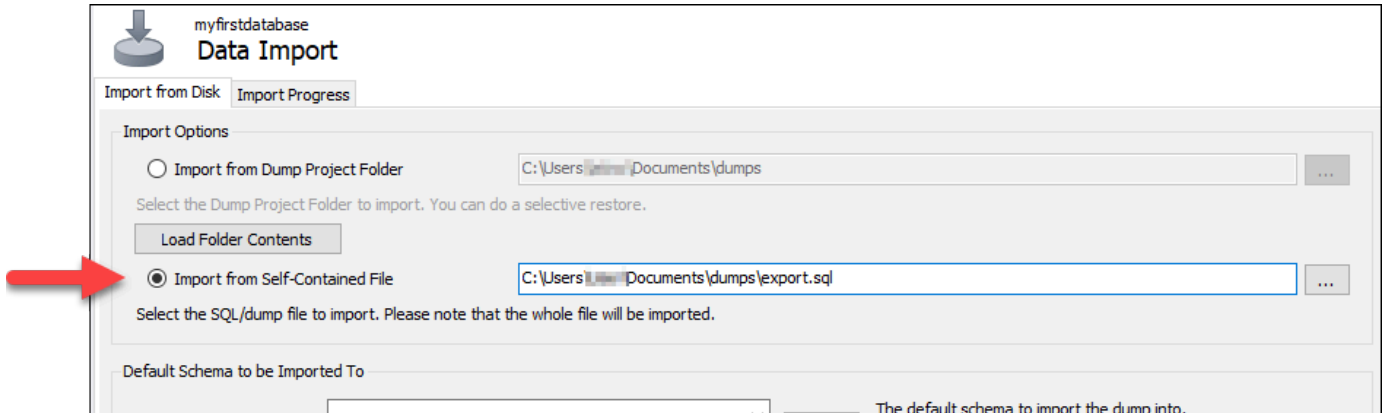
Note

To learn how to connect MySQL Workbench to your database, see [Connect to your MySQL database](#).

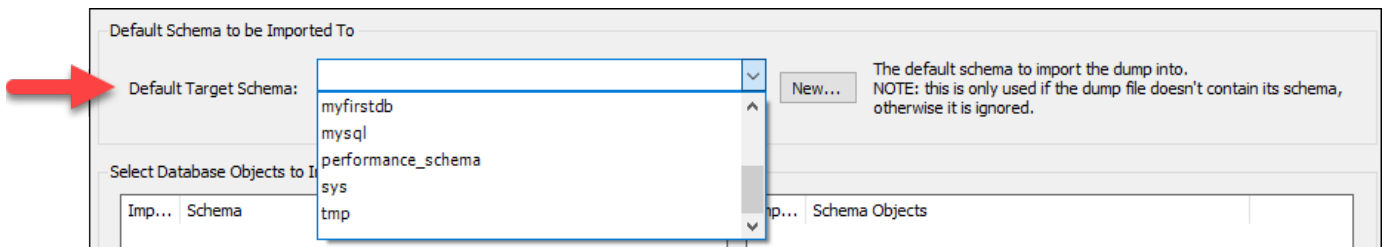
To import data into your database

1. Open MySQL Workbench.

- In the list of MySQL Connections, choose your MySQL managed database.
- Choose **Data Import/Restore** from the left-navigation menu.
- In the Data Import pane, choose **Import from Self-Contained File** under the **Import Options** section.

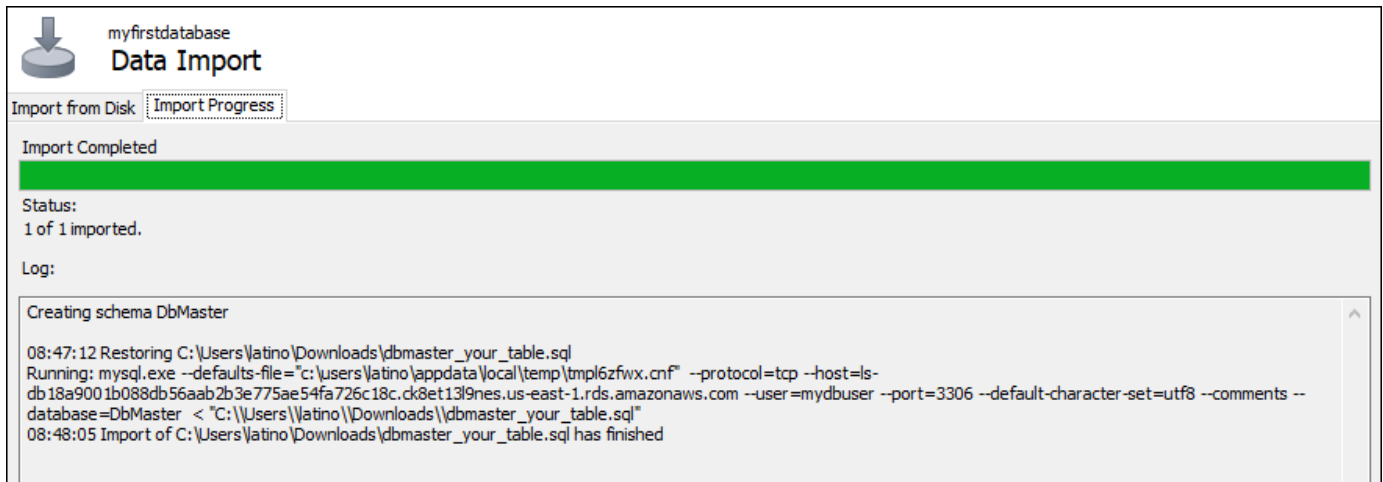


- Choose the ellipsis button to browse your local drive for the .SQL file that you want to import.
- Choose the .SQL file to import, then choose **Open**.
- Choose the **Default Target Schema** drop-down menu, then select the existing database to import the file to. You can also create a new database by choosing **New**.



- Choose **Start Import** to start the import.

Your import may take a few minutes or more depending on the size of the .SQL file. After the import is complete, you should see a message similar to the following:



Import data into your PostgreSQL database in Lightsail

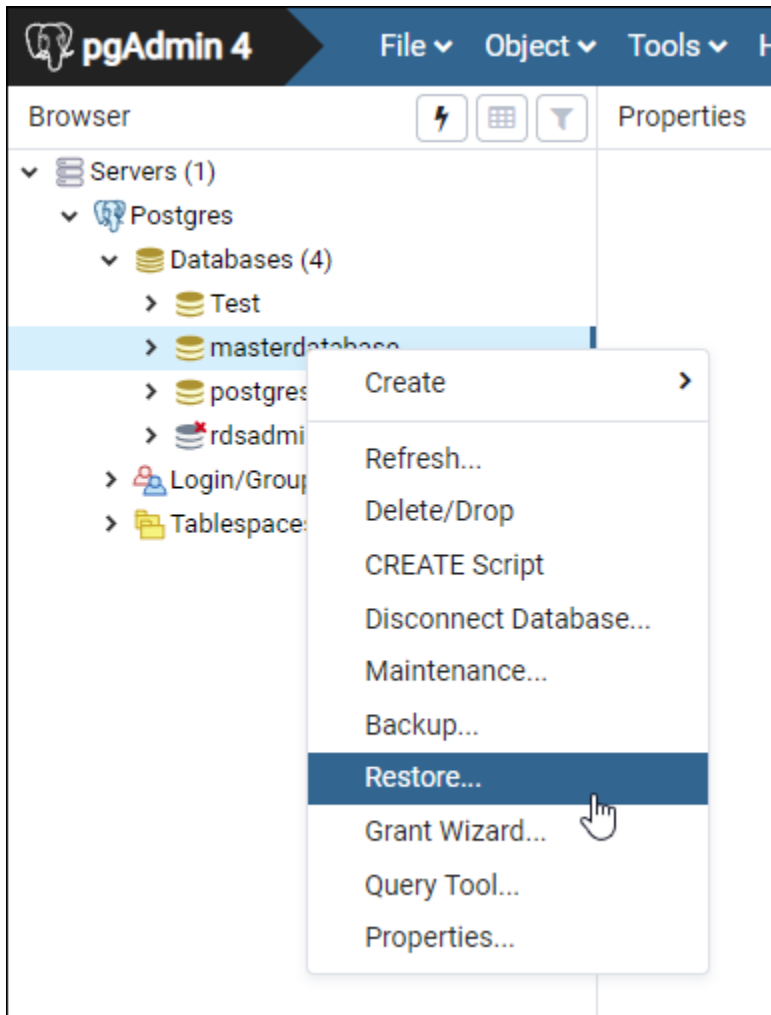
You can import a database backup file into your PostgreSQL managed database in Amazon Lightsail using pgAdmin.

Note

To learn how to connect pgAdmin to your database, see [Connect to your PostgreSQL database](#). For more information about creating a PostgreSQL database backup that you can import to another database, see [Backup Dialog](#) in the pgAdmin documentation.

To import a backup file into your database

1. Open pgAdmin.
2. In the list of server connections, double-click your PostgreSQL managed database in Amazon Lightsail to connect to it.
3. Expand the **Databases** node
4. Right-click the database in which you would like to import data from a database backup file, then choose **Restore**.

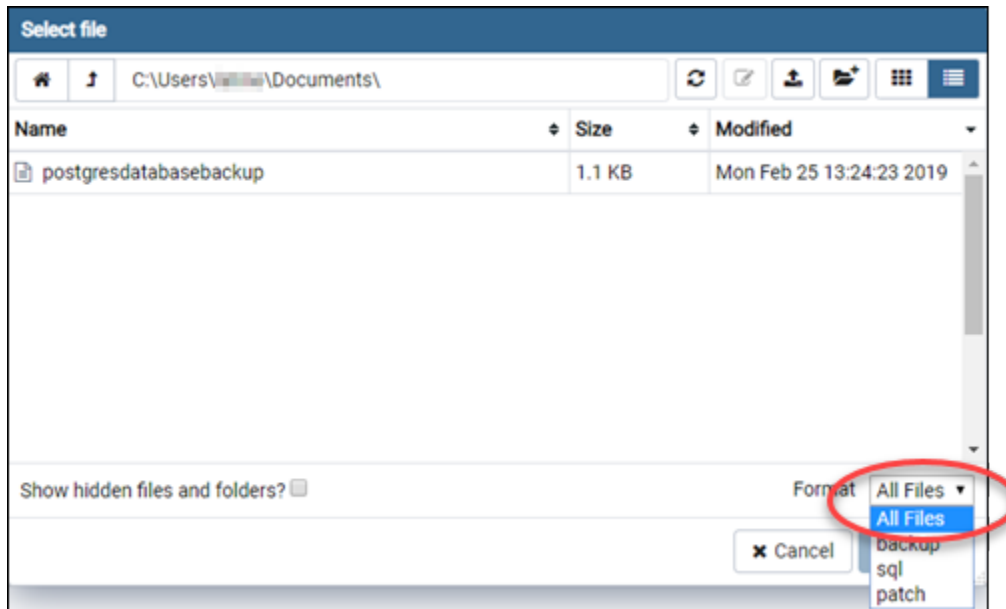


5. In the **Restore** form, complete the following fields:

- **Format** — Choose the format of your backup file.
- **Filename** — Choose the ellipsis icon, then locate and choose the database backup file on your local drive. After the file is highlighted, choose **Select** to go back to the **Restore** prompt.

Note

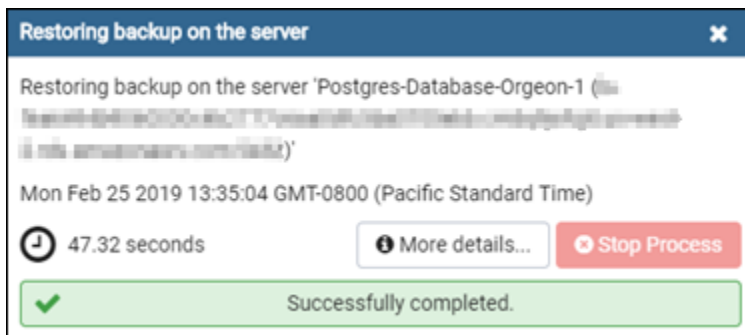
Choose the **Format** drop-down menu, and select **All files** to view all file formats on your local drive. Your backup file might be saved as a file type that is different than what is selected by default (sql).



- **Number of jobs** and **Role name** — Leave these fields blank.

6. Choose **Restore** to start the import.

Your import may take a few minutes or more depending on the size of the database backup file. After the import is complete, you should see a message similar to the following:



View your Lightsail database logs and history

View your database logs and history of changes in the Amazon Lightsail console. Database logs provide useful information that could help you diagnose issues with your database. Likewise, database history shows you changes made to your database, which allows you to associate problems with a recent change.

To view your database logs

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database for which you want to view logs.
4. Choose the **Logs and history** tab.

The page displays the database logs and history of changes made to your database.

5. Choose a database log. The following database logs are available:

MySQL database logs

- **Error log** — A record of mysqld start up and shutdown times. It also contains diagnostic messages such as errors, warnings, and notes that occur during server start up and shutdown, and while the server is running. For more information, see the error log article in the [MySQL 5.6](#), [MySQL 5.7](#), or [MySQL 8.0](#) documentation.
- **General log** — A general record of what mysqld is doing. The server writes information to this log when clients connect or disconnect, and logs each SQL statement received from clients. For more information, see the general query log article in the [MySQL 5.6](#), [MySQL 5.7](#), or [MySQL 8.0](#) documentation.
- **Slow query log** — A record of SQL statements that took more than long_query_time seconds to run, and required at least min_examined_row_limit rows to be examined. For more information, see the slow query log article in the [MySQL 5.6](#), [MySQL 5.7](#), or [MySQL 8.0](#) documentation.

Note

The general and slow query logs are disabled by default for MySQL databases. You can enable these logs, and begin collecting data, by updating a few database parameters. For more information, see [Enabling the MySQL database general and slow query logs in Amazon Lightsail](#).

PostgreSQL database logs

- **Postgres log** — A record of database start up and shutdown times. It can also contain diagnostics, such as errors, warnings, notices, and debug messages that occur during

database start up, shutdown, and while the database is running. For more information, see the error reporting and logging article in the [PostgreSQL 9.6](#), [PostgreSQL 10](#), [PostgreSQL 11](#), or [PostgreSQL 12](#) documentation.

Topics

- [Enable general and slow query logs for your Lightsail MySQL database](#)

Enable general and slow query logs for your Lightsail MySQL database

The [general and slow query logs](#) are disabled by default for MySQL databases in Amazon Lightsail. You can enable these logs, and begin collecting data, by updating a few database parameters. Update the database parameters by using the Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs. In this guide, we show you how to use the AWS CLI to update your database parameters and enable the general and slow query logs. We also provide additional options for controlling the general and slow query logs, and how log data retention is handled.

Prerequisite

If you haven't done so already, install and configure the AWS CLI. For more information, see [Configure the AWS Command Line Interface to work with Amazon Lightsail](#).

Enable the general and slow query logs in the Lightsail console

To enable the general and slow query logs in the Lightsail console, you must update the `general_log` and `slow_query_log` database parameters with a value of 1, and the `log_output` parameter with a value of FILE.

To enable the general and slow query logs in the Lightsail console

1. Open a Terminal or Command Prompt window.
2. Enter the following command to update the `general_log` parameter to a value of 1, which is true, or enabled.

```
aws lightsail update-relational-database-parameters --  
region Region --relational-database-name DatabaseName --parameters  
"parameterName=general_log,parameterValue=1,applyMethod=pending-reboot"
```

In the command, replace:

- *DatabaseName* with the name of your database.
 - *Region* with the AWS Region of your database.
3. Enter the following command to update the `slow_query_log` parameter to a value of 1, which is true, or enabled.

```
aws lightsail update-relational-database-parameters --  
region Region --relational-database-name DatabaseName --parameters  
"parameterName=slow_query_log,parameterValue=1,applyMethod=pending-reboot"
```

In the command, replace:

- *DatabaseName* with the name of your database.
 - *Region* with the AWS Region of your database.
4. Enter the following command to update the `log_output` parameter to a value of FILE, which writes the log data to a system file and enables it to be displayed in the Lightsail console.

```
aws lightsail update-relational-database-parameters --  
region Region --relational-database-name DatabaseName --parameters  
"parameterName=log_output,parameterValue=FILE,applyMethod=pending-reboot"
```

In the command, replace:


- *DatabaseName* with the name of your database.
 - *Region* with the AWS Region of your database.
5. Enter the following command to reboot the database and make the changes effective.

```
aws lightsail reboot-relational-database --region Region --relational-database-  
name DatabaseName
```

In the command, replace:

- *DatabaseName* with the name of your database.
- *Region* with the AWS Region of your database.

At this point, your database becomes unavailable while it reboots. Wait a few minutes, then sign in to the [Lightsail console](#) to view the general and slow query logs for your database. For more information, see [Viewing your database logs and history in Amazon Lightsail](#).


 **Note**

For more information about updating database parameters, see [Updating database parameters in Amazon Lightsail](#).

Control additional database log options

To control additional options for the MySQL general and slow query logs, update the following parameters:

- `log_output` — Set this parameter to `TABLE`. This writes general queries to the `mysql.general_log` table, and slow queries to the `mysql.slow_log` table. You can also set the `log_output` parameter to `NONE` to disable logging.

 **Note**

Setting the `log_output` parameter to `TABLE` disables the general and slow query log data from displaying in the Lightsail console. Instead, you must refer to the `mysql.general_log` and `mysql.slow_log` tables on your database to view the log data.

- `long_query_time` — To prevent fast-running queries from being logged in the slow query log, specify a value for the shortest query execution time to be logged, in seconds. The default is 10 seconds, and the minimum is 0. If the `log_output` parameter is set to `FILE`, you can specify a floating point value that goes to microsecond resolution. If the `log_output` parameter is set to `TABLE`, you must specify an integer value with second resolution. Only queries whose execution time exceeds the `long_query_time` parameter value are logged. For example, setting `long_query_time` to 0.1 prevents any query that runs for less than 100 milliseconds from being logged.

- `log_queries_not_using_indexes` — To log all queries that do not use an index to the slow query log, set to 1. The default is 0. Queries that do not use an index are logged even if their execution time is less than the value of the `long_query_time` parameter.

Log data retention

When logging is enabled, table logs are rotated, or log files are deleted, at regular intervals. This measure is a precaution to reduce the possibility of a large log file either blocking database use or affecting performance. When the `log_output` parameter is set to `FILE` or `TABLE`, logging is handled as follows:

- When `FILE` logging is enabled, log files are examined every hour and log files older than 24 hours are deleted. In some cases, the remaining combined log file size after the deletion might exceed the threshold of 2 percent of a database's allocated space. In these cases, the largest log files are deleted until the log file size no longer exceeds the threshold.
- When `TABLE` logging is enabled, log tables are rotated every 24 hours in some cases.

This rotation occurs if the space used by the table logs is more than 20 percent of the allocated storage space or the size of all logs combined is greater than 10 GB.

If the amount of space used for a database is greater than 90 percent of the database's allocated storage space, then the thresholds for log rotation are reduced.

Log tables are then rotated if the space used by the table logs is more than 10 percent of the allocated storage space or the size of all logs combined is greater than 5 GB.

You can subscribe to the `low_free_storage` event to be notified when log tables are rotated to free up space.

- When log tables are rotated, the current log table is copied to a backup log table and the entries in the current log table are removed. If the backup log table already exists, then it is deleted before the current log table is copied to the backup. You can query the backup log table. The backup log table for the `mysql.general_log` table is named `mysql.general_log_backup`. The backup log table for the `mysql.slow_log` table is named `mysql.slow_log_backup`.
- You can rotate the `mysql.general_log` table by calling the `mysql.rds_rotate_general_logprocedure`. You can rotate the `mysql.slow_log` table by calling the `mysql.rds_rotate_slow_logprocedure`.

- Table logs are rotated during a database version upgrade.

Create a snapshot of your Lightsail database

You can create a snapshot of your managed database in Amazon Lightsail. A snapshot is a copy of your database that you can use to restore it if something goes wrong. You can also use a snapshot to create a new database using a different plan, such as a high availability or standard plan.

When you create a snapshot of a standard database, the database becomes unavailable from a few seconds to a few minutes, depending on the size. High availability databases are not affected by snapshot operations because the snapshot is created using the standby database.

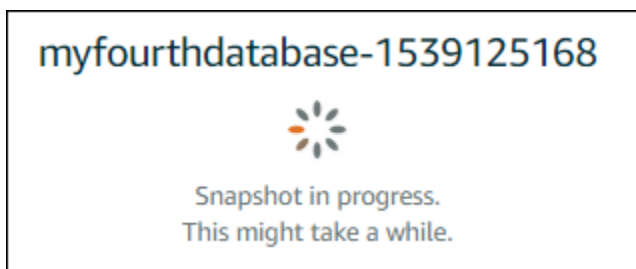
To create a snapshot of your database

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database for which you want to create a snapshot.
4. Choose the **Snapshots & restore** tab.
5. Under the **Manual snapshots** section of the page, choose **Create snapshot**, then enter a name for your snapshot.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
6. Choose **Create**.

The snapshot creation process begins and a status of **Snapshot in progress** is shown.



After the snapshot creation process is complete, the new snapshot is listed under the **Recent snapshots** section. You can also view all of the snapshots for your account in the Lightsail home page, under the **Snapshots** tab.



Next steps

After your snapshot is ready, you can create a new database from the snapshot, which is a duplicate of the original database. For more information, see [Create a database from a snapshot](#).

Topics

- [Create a database from a point-in-time backup in Amazon Lightsail](#)
- [Create a database from a snapshot in Lightsail](#)

Create a database from a point-in-time backup in Amazon Lightsail

You can create a new managed database by using a point-in-time backup in Amazon Lightsail. Point-in-time backups of your database are available in 5-minute increments, and for the previous seven days. This gives you the ability to restore a failed database to a specific date and time in the last week.

You can also create a new database from a snapshot. For more information, see [Creating a database from a snapshot in Amazon Lightsail](#).


To create a database from a point-in-time backup

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database for which you want to change plans.
4. Choose the **Snapshots and restore** tab.

5. Under the **Emergency restore** section, select the date and time of the backup that you want to use for your new database.

Emergency restore

Lightsail retains a week of minute-to-minute backups of your database. Select a point in time from the last week to create a new database from that backup.


 If you recently enabled data import mode, you can only restore from a point in time after you disabled it.

Today ▼ , 17 ▼ : 50 ▼ — Pacific Daylight Time (GMT-7) ▼

Restore to new database

6. Choose **Restore to new database**.
7. On the **Create a new database** page, choose **Change zone** to select a different Availability Zone. Your new database is then created in the same AWS Region as the snapshot that you selected earlier.
8. Choose your new database plan.

Pick a high availability or a standard database plan. A database created with a high availability plan has a primary database and a secondary standby database in another Availability Zone for failover support. For more information, see [High availability databases](#).

 **Note**

You cannot choose a database plan that is smaller than the plan of the original database.

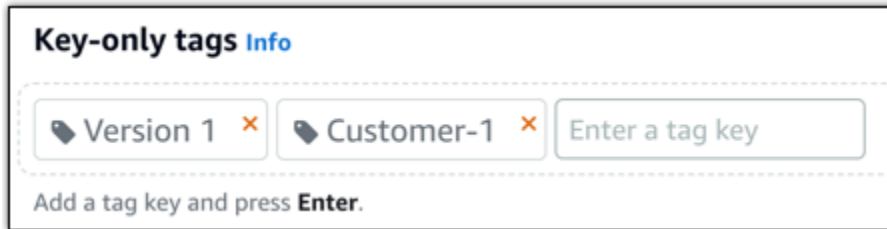
9. Enter a name for your database.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

10. Choose one of the following options to add tags to your database:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



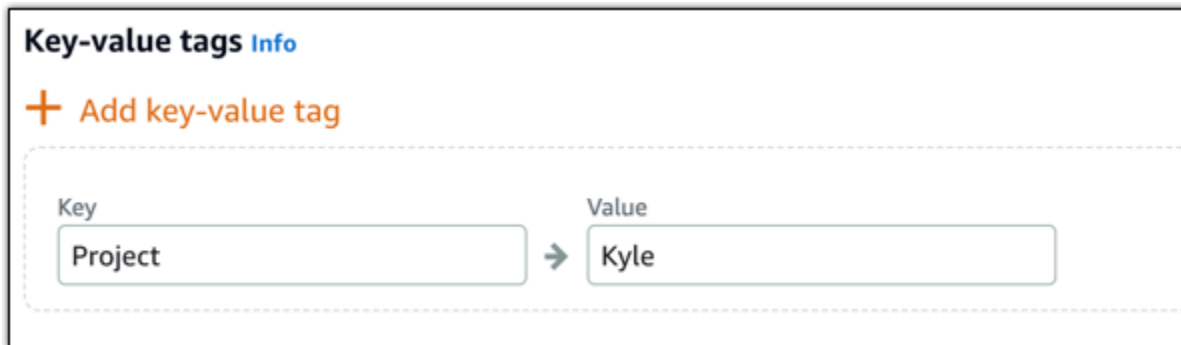
Key-only tags Info

Version 1 × Customer-1 × Enter a tag key

Add a tag key and press **Enter**.

- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key Value

Project → Kyle

Note

For more information about key-only and key-value tags, see [Tags](#).

11. Choose **Create database**.

Within minutes, your new Lightsail database is ready with the new database plan or bundle.

Next steps

Complete the following actions after your new database is up and running:

- Delete the original database if you no longer need it. For more information, see [Delete your database](#).
- Databases created from a point-in-time backup are configured to use a strong password created by Lightsail. For more information, see [Manage your database password](#).

Create a database from a snapshot in Lightsail

You can create a new managed database from a snapshot in Amazon Lightsail if something goes wrong with your original database. You can also change your database to a different plan, such as a high availability or standard plan. You can also create a new database from a point-in-time backup of your original database. For more information, see [Create a database from a point-in-time backup in Amazon Lightsail](#).

When you create the duplicate database, you can choose a different or larger plan than the original database. However, you can't choose a smaller plan than the original database.

Note

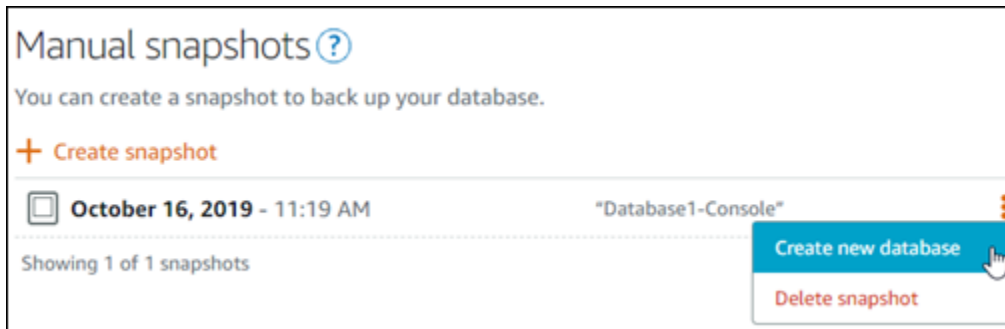
A database created with a high availability plan has a primary database and a secondary standby database in another Availability Zone for failover support. For more information, see [High availability databases](#).

To create a database from a snapshot

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database that you want to duplicate by creating a new database from a snapshot.
4. Choose the **Snapshots & restore** tab.
5. Under the **Manual snapshots** section of the page, choose the actions menu icon (:) next to the snapshot from which you want to create a new database, and choose **Create new database**.

Note

You need a snapshot of your database to work from. If you haven't created a snapshot yet, see [Create a snapshot of your database](#).



6. Choose **Create new database**.
7. On the **Create a new database** page, choose **Change zone** to select a different Availability Zone. Your new database is created in the same AWS Region as the snapshot that you selected earlier.
8. Choose your new database plan.

Select a high availability or a standard database plan. A database created with a high availability plan has a primary database and a secondary standby database in another Availability Zone for failover support. For more information, see [High availability databases](#).

Note

You cannot choose a database plan that is smaller than the plan of the original database used to create the snapshot.

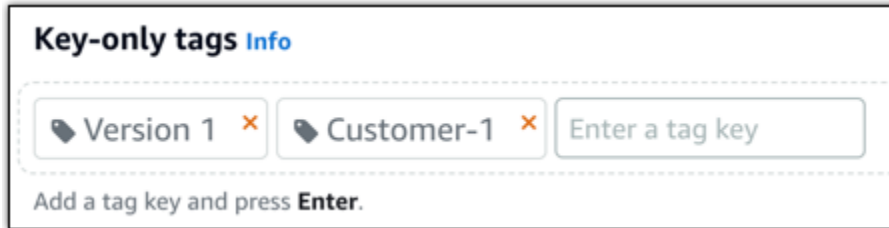
9. Enter a name for your database.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

10. Choose one of the following options to add tags to your database:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



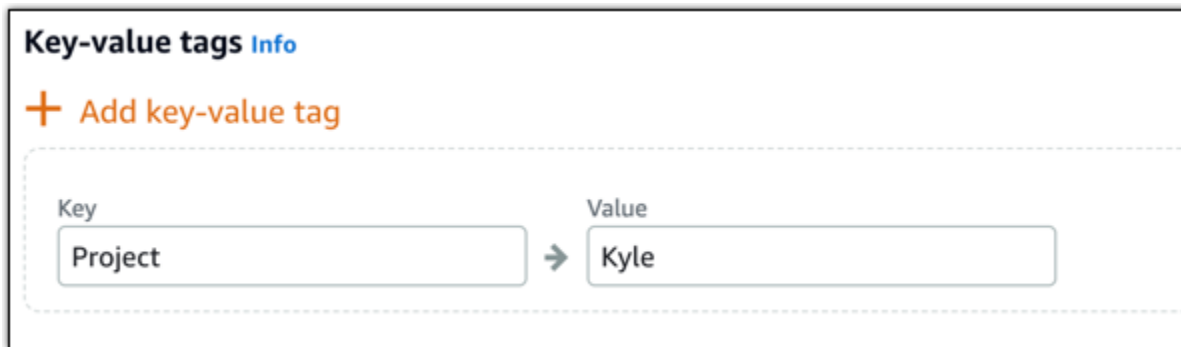
Key-only tags Info

Version 1 × Customer-1 × Enter a tag key

Add a tag key and press **Enter**.

- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key Value

Project → Kyle

Note

For more information about key-only and key-value tags, see [Tags](#).

11. Choose **Create database**.

Within minutes, your new Lightsail database is ready with the new database plan or bundle.

Next steps

Complete the following actions after your new database is up and running:

- If you're creating a new database to replace an existing database, and you have an application that depends on the existing database, then make sure to update your application dependencies to your new database.
- Delete the original database if you no longer need it. For more information, see [Delete your database](#).
- Databases created from a snapshot are configured to use a strong password created by Lightsail. For more information, see [Manage your database password](#).

Download an SSL certificate for your Lightsail managed database

You can use Secure Socket Layer (SSL) or Transport Layer Security (TLS) from your application to encrypt a connection to a managed database in Amazon Lightsail running MySQL, or PostgreSQL. Each DB engine has its own process for implementing SSL/TLS. For more information, see [Using SSL to connect to your MySQL database](#) or [Using SSL to connect to your PostgreSQL database](#).

Note

The certificates available for download are labeled for Amazon Relational Database Service (Amazon RDS), but also work for managed databases in Lightsail.

Certificate bundles for all AWS Regions

To get a certificate bundle that contains both the intermediate and root certificates for all AWS Regions, or if your application is on Microsoft Windows and requires a PKCS7 file, see [Certificate bundles for all AWS Regions](#) in the Amazon Relational Database Service User Guide.

This root certificate is a trusted root entity and should work in most cases. However, it might fail if your application doesn't accept certificate chains. If your application doesn't accept certificate chains, continue to the next section of this document.

Certificate bundles for specific AWS Regions

To get a certificate bundle that contains both the intermediate and root certificates for a specific AWS Region, see [Certificate bundles for specific AWS Regions](#) in the Amazon Relational Database Service User Guide.

Update the CA certificate version for your Lightsail database

Amazon Lightsail has published new Certificate Authority (CA) certificates for connecting to your managed database using SSL/TLS. This guide describes how to upgrade to the new CA certificate. You can upgrade the certificate only by using the [update-relational-database](#) API action. The new certificates are referred to as `rds-ca-rsa2048-g1`, `rds-ca-rsa4096-g1`, and `rds-ca-ecc384-g1`. The old certificate is referred to as `rds-ca-2019`. We provide the CA certificates as an AWS security best practice. For information about the CA certificates for your managed database, and the supported AWS Regions, see [Downloading an SSL certificate for your managed database](#).

The old CA certificate (`rds-ca-2019`) expires on August 22, 2024. Therefore, we strongly recommend completing the steps in this guide as soon as possible to modify your managed database to use the new certificate. If your applications do not connect to your Lightsail managed database using SSL/TLS, no action is required. If these steps are not completed, your applications will fail to connect to your managed database using SSL/TLS after August 22, 2024.

New managed databases created after January 26, 2024 will use the `rds-ca-rsa2048-g1` certificate by default. If you want to temporarily modify new managed databases to use the old certificate (`rds-ca-2019`), you can do so using the AWS Command Line Interface (AWS CLI). Any managed databases created prior to January 26, 2024 uses the `rds-ca-2019` certificate until you update them to the `rds-ca-rsa2048-g1`, `rds-ca-rsa4096-g1`, and `rds-ca-ecc384-g1` certificates.

Note

Test the steps in this guide on a development or staging environment before using them on your production environments.

Prerequisites

- In this guide, you will use AWS CloudShell to perform the upgrade. CloudShell is a browser-based, pre-authenticated shell that you can launch directly from the Lightsail console. With CloudShell, you can run AWS Command Line Interface (AWS CLI) commands using your preferred shell, such as Bash, PowerShell, or Z shell. You can do this without downloading or installing command line tools. For more information about how to set up and use CloudShell, see [AWS CloudShell in Lightsail](#).

- Before completing the following steps, make sure to update your database applications to use the new SSL/TLS certificate. The methods for updating applications for new SSL/TLS certificates depend on your specific applications. Work with your application developers to update the SSL/TLS certificates for your applications. To learn more about updating applications for new SSL/TLS certificates, see [Updating Applications to Connect to MySQL DB Instances Using New SSL/TLS Certificates](#) or [Updating Applications to Connect to PostgreSQL DB Instances Using New SSL/TLS Certificates](#) in the *Amazon Relational Database Service User Guide*.

Identify the active CA certificate for your managed database

Complete the following steps to identify the active CA certificate for your Lightsail database instance.

1. Open a Terminal, [AWS CloudShell](#), or Command Prompt window.
2. Enter the following command to identify the active CA certificate for your managed database.

```
aws lightsail get-relational-database --relational-database-name DatabaseName --  
region DatabaseRegion | grep "caCertificateIdentifier"
```

In the command, replace *DatabaseName* with the name of the database you want to modify, and *DatabaseRegion* with the AWS Region that the database instance is in.

Example

```
aws lightsail get-relational-database --relational-database-name Database-1 --  
region us-east-1 | grep "caCertificateIdentifier"
```

The command will return the ID of the active CA certificate for your database.

Example

```
"caCertificateIdentifier": "rds-ca-rsa2048-g1"
```

Modify your managed database to use the new CA certificate

Complete the following steps to modify your managed database in Lightsail to use one of the new CA certificates (`rds-ca-rsa2048-g1`, `rds-ca-rsa4096-g1`, and `rds-ca-ecc384-g1`).

1. Open a Terminal, [AWS CloudShell](#), or Command Prompt window.
2. Enter the following command to use the new certificate on your managed database.

```
aws lightsail update-relational-database --relational-database-name DatabaseName --ca-certificate-identifier rds-ca-rsa2048-g1
```

In the command, replace *DatabaseName* with the name of the database you want to modify.

Example

```
aws lightsail update-relational-database --relational-database-name Database-1 --ca-certificate-identifier rds-ca-rsa2048-g1
```

The CA certificate used by your managed database will be updated during your database's next maintenance window, or immediately if you add the `--apply-immediately` parameter to the end of the command.

Modify your managed database to use the old CA certificate

Complete the following steps to modify your managed database in Lightsail to use the old CA certificate (`rds-ca-2019`). Do this only if you experience a critical issue with one of the new certificates (`rds-ca-rsa2048-g1`, `rds-ca-rsa4096-g1`, and `rds-ca-ecc384-g1`) and need to temporarily revert the old one.

1. Open a Terminal, [AWS CloudShell](#), or Command Prompt window.
2. Enter the following command to use the `rds-ca-2019` on your managed database.

```
aws lightsail update-relational-database --relational-database-name DatabaseName --ca-certificate-identifier rds-ca-2019
```

In the command, replace *DatabaseName* with the name of the database you want to modify.

Example

```
aws lightsail update-relational-database --relational-database-name Database-1 --ca-certificate-identifier rds-ca-2019
```

The CA certificate used by your managed database will be updated during your database's next maintenance window, or immediately if you add the `--apply-immediately` parameter to the end of the command.

Change the preferred maintenance and backup windows for your Lightsail database

When a new version of a database is supported by Amazon Lightsail, your existing managed database can be upgraded to it. There are two kinds of upgrades—minor version upgrades and major version upgrades. Currently, Lightsail supports only minor version upgrades.

Minor version upgrades, and other database maintenance tasks, are performed automatically during the preferred maintenance window for your database. The preferred maintenance window is a 30-minute window selected at random from an 8-hour block of time for each AWS Region. It occurs on a random day of the week. Database backups are performed during the preferred backup window. The preferred backup window is a 30-minute window selected at random from an 8-hour block of time for each AWS Region. It also occurs on a random day of the week.

Note

For more information about the preferred maintenance window time blocks for each region, see the [Maintaining a DB Instance](#) guide in the Amazon Relational Database Service (Amazon RDS) documentation. For more information about the preferred backup window time blocks for each region, see the [Working With Backups](#) guide in the Amazon RDS documentation.

This guide shows you how to change the preferred maintenance and backup windows, so that they occur when your database is under its lowest load.

Prerequisites

You must use the AWS Command Line Interface (AWS CLI) to change your database preferred maintenance and backup windows.

Complete the following prerequisites:

- **Install the AWS CLI** — For more information, see [Installing the AWS CLI](#).
- **Configure the AWS CLI** — For more information, see [Configuring the AWS CLI](#).

Change your database maintenance window

Your database may become unavailable during maintenance or backup operations. Therefore, you may want to change your preferred maintenance or backup window to a time in which your database is under its lowest load.

To change your database maintenance window

1. Open a Terminal or Command Prompt window.
2. Enter the following command to get the name of the database for which you want to change the maintenance window:

```
aws lightsail get-relational-databases
```

You should see a result similar to the following example:


```
{
  "relationalDatabases": [
    {
      "name": "myfirsttestdatabase",
      "arn": "arn:aws:lightsail:us-east-1:13869536:lightsail:relationalDatabase:mysql-084884343714-084884343714-084884343714",
      "supportCode": "084884343714/ls-8e39329c39ee",
      "createdAt": 1538755937.532,
      "location": {
        "availabilityZone": "us-east-1a",
        "regionName": "us-east-1"
      },
      "resourceType": "RelationalDatabase",
      "relationalDatabaseBlueprintId": "mysql_5_7",
      "relationalDatabaseBundleId": "medium_1_0",
      "masterDatabaseName": "myseconddb",
      "hardware": {
        "cpuCount": 2,
        "diskSizeInGb": 120,
        "ramSizeInGb": 4.0
      },
      "state": "available",
      "backupRetentionEnabled": false,
      "pendingModifiedValues": {},
      "engine": "mysql",
      "engineVersion": "5.7.23",
      "masterUsername": "myfirstuser",
      "parameterApplyStatus": "in-sync",
      "preferredBackupWindow": "08:49-09:19",
      "preferredMaintenanceWindow": "mon:10:16-mon:10:46",
      "publiclyAccessible": true,
      "masterEndpoint": {
        "port": 3306,
        "address": "11-8a190319e269ec346e5fa11a25e5e29097cd4f04.us-east-1.rds.amazonaws.com"
      },
      "pendingMaintenanceActions": []
    }
  ]
}
```

Note

If the database that you want to modify is not listed, confirm that your AWS CLI is configured for the AWS Region where the database is located. For more information, see [Configure the AWS CLI](#).

- Highlight the name of the database that you want to modify and press **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using macOS, to copy it to your clipboard so that you can use it in the next step.

```
{
  "relationalDatabases": [
    {
      "name": "myfirsttestdatabase",
      "arn": "arn:aws:lightsail:us-east-1:13869536",
      "supportCode": "084884343714/ls-8e39329c39ee",
      "createdAt": 1538755937.532,
      "location": {
```

- Enter one of the following commands depending on the preferred window that you are changing.

- Enter the following command to change the database maintenance window.

```
aws lightsail update-relational-database --relational-database-name DatabaseName
--preferred-maintenance-window MaintenanceWindow
```

In the command, replace:

- *DatabaseName* with the name of the database.
- *MaintenanceWindow* with the new maintenance window time frame.

Define the preferred maintenance window time in ddd:hh24:mi-ddd:hh24:mi format. It also must be in Universal Coordinated Time (UTC) format, and defined for a minimum window of 30 minutes. The preferred maintenance window cannot overlap the preferred backup window.

Example:

```
aws lightsail update-relational-database --relational-database-
name myproductiondb --preferred-maintenance-window Tue:16:00-Tue:16:30
```

- Enter the following command to change the database backup window.

```
aws lightsail update-relational-database --relational-database-name DatabaseName
--preferred-backup-window BackupWindow
```

In the command, replace:

- *DatabaseName* with the name of the database.
- *BackupWindow* with the new backup window time frame.

Define the preferred backup window time in hh24:mi-hh24:mi format. It also must be in Universal Coordinated Time (UTC) format, and defined for a minimum window of 30 minutes. The preferred backup window cannot overlap the preferred maintenance window.

Example:

```
aws lightsail update-relational-database --relational-database-
name myproductiondb --preferred-backup-window 14:00-14:30
```

You should see a result similar to the following example:

```
{
  "operations": [
    {
      "id": "xxxxxxxx-xxxx-4xxx-xxxx-xxxxxxxxxxxx",
      "resourceName": "myfirsttestdatabase",
      "resourceType": "RelationalDatabase",
      "createdAt": 1539124310.116,
      "location": {
        "availabilityZone": "us-east-1a",
        "regionName": "us-east-1"
      },
      "isTerminal": true,
      "operationType": "UpdateRelationalDatabase",
      "status": "Succeeded",
      "statusChangedAt": 1539124310.283
    }
  ]
}
```

Next steps

Here are a few guides to help you manage your database:

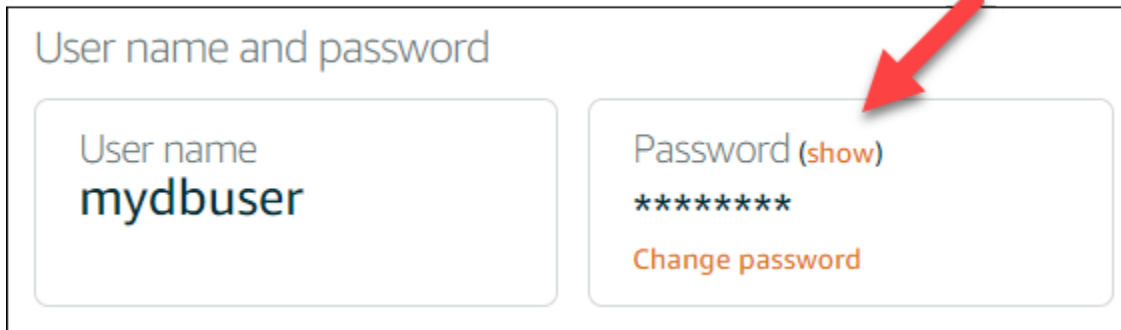
- [Configure the data import mode for your database](#)
- [Configure the public mode for your database](#)
- [Manage your database password](#)
- [Connect to your MySQL database](#)
- [Connect to your PostgreSQL database](#)
- [Import data into your MySQL database](#)
- [Import data into your PostgreSQL database](#)
- [Create a snapshot of your database](#)

Manage your Lightsail database password

When you create a new database in Amazon Lightsail, you can let Lightsail create a strong password for you or specify your own. You can view or change the current database password at any time in the Lightsail console.

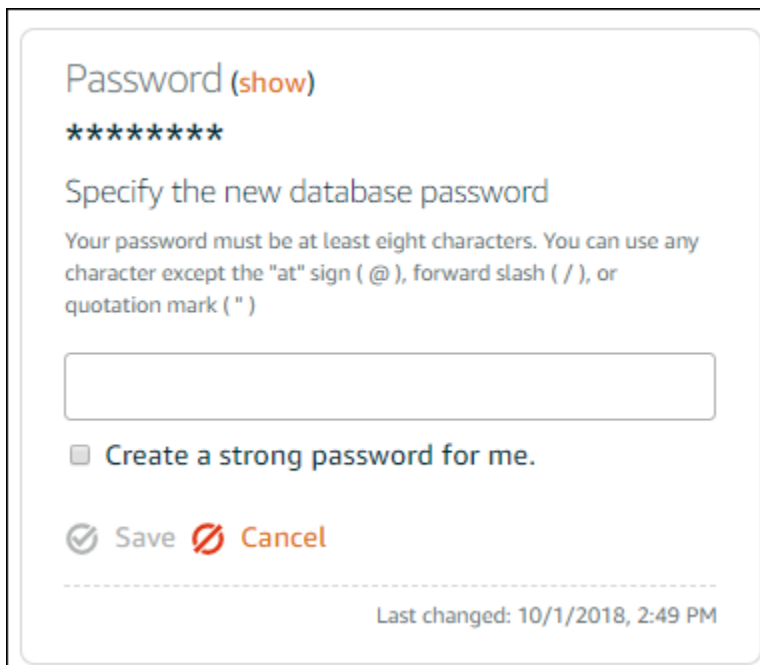
To manage your database password

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database for which you want to manage the password.
4. On the **Connect** tab, under the **User name and passwords** section, choose **Show** to view the current database password.



5. To change the database password, choose **Change password**.

You can opt to have Lightsail create a strong password for you, or you can enter your own password into the text box. The password can include any printable ASCII character except "/", "", or "@". For MySQL databases, the password must contain from 8 to 41 characters. For PostgreSQL, the password must contain from 8 to 128 characters.



6. Choose **Save** when you're done.

A database password change is applied immediately. If you entered your own password, the password is saved immediately. If Lightsail created the password for you, it is generated within a few seconds. Choose **Show** to view the new password.

Next steps

Here are a few guides to help you manage your database in Lightsail:

- [Connect to your MySQL database](#)
- [Connect to your PostgreSQL database](#)
- [Create a snapshot of your database](#)

Configure public mode for your Lightsail database

Your managed database in Amazon Lightsail is accessible only by your Lightsail resources (instances, load balancers, etc.) that are in the same Lightsail account. One common scenario is to create both a Lightsail instance with a public-facing web application and a Lightsail database that is not publicly accessible, and then connect the two.

Enable the public mode feature to make your database publicly accessible. This way, anyone with the database endpoint, port, user name, and password can connect to your database. For more information, see [Connect to your MySQL database](#) or [Connect to your PostgreSQL database](#).

To configure the public mode for your database

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database for which you want to configure public mode.
4. Choose the **Networking** tab.
5. Under the **Public mode** section, use the toggle to turn it on. Likewise, use the toggle to turn it off.

Public mode

When public mode is enabled, anyone with your database user name and password can connect to it. When this mode is disabled, only your Lightsail resources in the same Region as your database can connect to it



Public mode is **disabled**.

Only your Lightsail resources in the same Region as your database can connect to it.

The public accessibility setting begins applying immediately but may require a few minutes to complete. During this time, the status of your database changes to **Modifying**. The status of your database changes to **Available** after the public accessibility setting is applied.

Next steps

Here are a few guides to help you manage your database:

- [Configure the data import mode for your database](#)
- [Manage your database password](#)
- [Connect to your MySQL database](#)
- [Connect to your PostgreSQL database](#)
- [Import data into your MySQL database](#)
- [Import data into your PostgreSQL database](#)
- [Create a snapshot of your database](#)

Update Lightsail database parameters

Database parameters, also known as database system variables, define fundamental properties of a managed database in Amazon Lightsail. For example, you can define a database parameter to limit the number of database connections, or define another parameter to limit the database buffer pool size. This guide shows you how to get a list of the parameters for your managed database, and how to update them using the AWS Command Line Interface (AWS CLI).

Note

For more information about MySQL system variables, refer to the [MySQL 5.6](#), [MySQL 5.7](#), or [MySQL 8.0](#) documentation. For more information about PostgreSQL system variables, refer to the [PostgreSQL 9.6](#), [PostgreSQL 10](#), [PostgreSQL 11](#), or [PostgreSQL 12](#) documentation.

Prerequisites

- If you haven't done so already, install and configure the AWS CLI. For more information, see [Configure the AWS CLI to work with Lightsail](#).

Get a list of available database parameters

The database parameters differ depending on the database engine; therefore, you should get a list of the parameters available for your managed database. This will allow you to decide which parameter you want to modify, and the way in which that parameter becomes effective.

To get a list of available database parameters

1. Open a Terminal or Command Prompt window.
2. Enter the following command to get a list of parameters for your database..

```
aws lightsail get-relational-database-parameters --relational-database-name DatabaseName
```

In the command, replace *DatabaseName* with the name of your database.

You should see a result similar to the following example:

```

{
  "parameters": [
    {
      "allowedValues": "0,1",
      "applyMethod": "pending-reboot",
      "applyType": "static",
      "dataType": "boolean",
      "description": "Controls whether user-defined functions that have only an xxx symbol for the main
function can be loaded",
      "isModifiable": false,
      "parameterName": "allow-suspicious-udfs"
    },
    {
      "allowedValues": "0,1",
      "applyMethod": "pending-reboot",
      "applyType": "static",
      "dataType": "boolean",
      "description": "Controls whether the server autogenerates SSL key and certificate files in the dat
a directory, if they do not already exist.",
      "isModifiable": false,
      "parameterName": "auto_generate_certs"
    },
    {
      "allowedValues": "1-65535",
      "applyMethod": "pending-reboot",
      "applyType": "dynamic",
      "dataType": "integer",
      "description": "Intended for use with master-to-master replication, and can be used to control the
operation of AUTO_INCREMENT columns",
      "isModifiable": true,
      "parameterName": "auto_increment_increment"
    },
    {
      "allowedValues": "1-65535",
      "applyMethod": "pending-reboot"
    }
  ]
}

```

Note

A next page token ID is listed if the parameter results are paginated. Make note of the next page token ID and use it as shown in the next step to view the next page of parameter results.

3. If your results are paginated, use the following command to view the additional set of parameters. Otherwise, skip to the next step.

```
aws lightsail get-relational-database-parameters --relational-database-
name DatabaseName --page-token NextPageTokenID
```

In the command, replace:

- *DatabaseName* with the name of your database.
- *NextPageTokenID* with the next page token ID.

The result displays the following information for each database parameter:

- **Allowed values** — Specifies the valid range of values for the parameter.

- **Apply method** — Specifies when the parameter change is applied. Allowed options are `immediate` or `pending-reboot`. See the following apply type for more information about how to define the apply method.
 - **Apply type** — Specifies the engine-specific submission type. If `dynamic` is listed, the parameter can be applied with an `immediate` apply method and the database will begin using the new parameter value immediately. If `static` is listed, the parameter can only be applied with a `pending-reboot` apply method and the database will begin using the new parameter only after it's restarted.
 - **Data type** — Specifies the valid data type for the parameter.
 - **Description** — Provides a description of the parameter.
 - **Is modifiable** — A Boolean value indicating whether the parameter can be modified. If `true` is listed, then the parameter can be modified.
 - **Parameter name** — Specifies the name of the parameter. Use this value together with the `update relational database operation` and the `parameter name` parameter.
4. Find the parameter you want to change, and make note of the parameter name, allowed values, and apply method. We recommend copying the parameter name to your clipboard to avoid entering it incorrectly. To do that, highlight the parameter name and press **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using macOS, to copy it to your clipboard. Then, press **Ctrl+V** or **Cmd+V** as appropriate to paste it.

After you identify the name of the parameter that you want to modify, continue to the next section of this guide to change the parameter to your desired value.

Update your database parameters

After you have the name of the parameter you want to change, perform the following steps to modify the parameter for your managed database in Lightsail:

To update your database parameters

- Enter the following command into a terminal or command prompt window to update a parameter for your managed database.

```
aws lightsail update-relational-database-parameters
--relational-database-name DatabaseName --parameters
"parameterName=ParameterName,parameterValue=NewParameterValue,applyMethod=ApplyMethod"
```

In the command, replace:

- *DatabaseName* with the name of your database.
- *ParameterName* with the name of the parameter you want to modify.
- *NewParameterValue* with the new value of the parameter.
- *ApplyMethod* with the apply method for the parameter.

If the parameter's apply type is `dynamic`, the parameter can be applied with an `immediate` apply method and the database will begin using the new parameter value immediately.

However, if the parameter apply type is `static`, the parameter can only be applied with a `pending-reboot` apply method and the database will begin using the new parameter only after it's restarted.

You should see a result similar to the following example:

```
{
  "operations": [
    {
      "id": "2c650987-11e8-463f-94d5-0c15aacaf12b",
      "resourceName": "myfirsttestdatabase",
      "resourceType": "RelationalDatabase",
      "createdAt": 1539204831.214,
      "location": {
        "availabilityZone": "us-east-1a",
        "regionName": "us-east-1"
      },
      "isTerminal": true,
      "operationType": "UpdateRelationalDatabaseParameters",
      "status": "Succeeded",
      "statusChangedAt": 1539204831.214
    }
  ]
}
```

The database parameter is updated depending on the apply method used.

Upgrade the major version of a Lightsail database

When Amazon Lightsail supports a new version of a database engine, you can upgrade your database to the new version. Lightsail offers two database blueprints, MySQL and PostgreSQL. This guide describes how to upgrade the major version for your MySQL or PostgreSQL database

instance. You can upgrade the database major version only by using the [update-relational-database](#) API action.

We will use AWS CloudShell to perform the upgrade. CloudShell is a browser-based, pre-authenticated shell that you can launch directly from the Lightsail console. With CloudShell, you can run AWS Command Line Interface (AWS CLI) commands using your preferred shell, such as Bash, PowerShell, or Z shell. You can do this without downloading or installing command line tools. For more information about how to set up and use CloudShell, see [AWS CloudShell in Lightsail](#).

Understand the changes

Major version upgrades can introduce a number of incompatibilities with the previous version. These incompatibilities can cause problems during an upgrade. You might need to prepare your database for the upgrade to be successful. For information about upgrading major versions of a database, see the following topics on the MySQL and PostgreSQL websites.

- [Preparing Your Installation for Upgrade](#)
- [MySQL Upgrade Checker Utility](#)
- [Upgrading a PostgreSQL Cluster](#)

Prerequisites

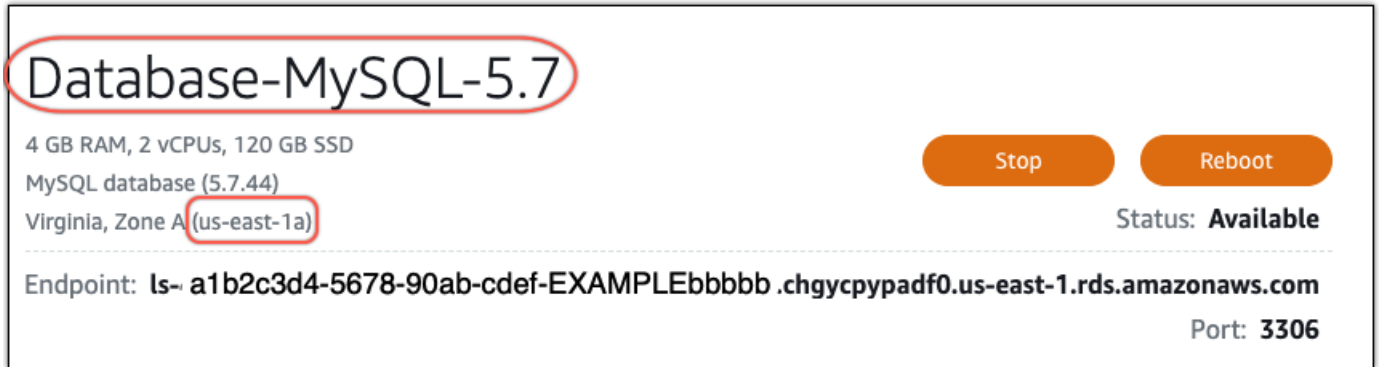
1. Verify that your application supports both major versions of the database.
2. We recommend that you create a snapshot of your database instance before making any changes. For more information, see [Create a snapshot of your Lightsail database](#).
3. (Optional) Create a new database instance from the snapshot that you just created. Because database updates require downtime, you can test the upgrade on the new database before you upgrade the database that's currently active. For more information about making a copy of your database, see [Create a snapshot of your Lightsail database](#).

Update the database major version

Lightsail supports major version upgrades for MySQL and PostgreSQL database instances. A MySQL database is used as an example in the following procedure. However, the process and commands are the same for a PostgreSQL database.

Complete the following procedure to upgrade the database major version for your Lightsail database.

1. Sign in to the [Lightsail console](#).
2. In the left navigation pane, choose **Databases**.
3. Note of the name and AWS Region for the database instance that you want to upgrade.



Database-MySQL-5.7

4 GB RAM, 2 vCPUs, 120 GB SSD

MySQL database (5.7.44)

Virginia, Zone A (us-east-1a)

Stop Reboot

Status: **Available**

Endpoint: ls-a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb .chgycpypadf0.us-east-1.rds.amazonaws.com

Port: **3306**

4. In the lower left corner of the Lightsail console, choose **CloudShell**. A CloudShell terminal will open in the same browser tab. When the command prompt displays, the shell is ready for interaction.
5. Enter the following command at the CloudShell prompt to get a list of database blueprint IDs that are available.

```
aws lightsail get-relational-database-blueprints
```

6. Note of the blueprint ID for the major version that you're upgrading to. For example, `mysql_8_0`.

```

AWS CloudShell
us-west-2

[cloudshell-user@ip-10-170-117 ~]$ aws lightsail get-relational-database-blueprints
{
  "blueprints": [
    {
      "blueprintId": "mysql_5_7",
      "engine": "mysql",
      "engineVersion": "5.7.44",
      "engineDescription": "MySQL Community Edition",
      "engineVersionDescription": "MySQL 5.7.44",
      "isEngineDefault": false
    },
    {
      "blueprintId": "mysql_8_0",
      "engine": "mysql",
      "engineVersion": "8.0.36",
      "engineDescription": "MySQL Community Edition",
      "engineVersionDescription": "MySQL 8.0.36",
      "isEngineDefault": true
    }
  ],
}

```

7. Enter the following command to upgrade the major version of your database. The upgrade will take place during the next maintenance window for your database. In the command, replace *DatabaseName* with the name of your database, *blueprintId* with the blueprint id of the major version that you are upgrading to, and *DatabaseRegion* with the AWS Region that your database is in.

```

aws lightsail update-relational-database \
  --relational-database-name DatabaseName \
  --relational-database-blueprint-id blueprintId \
  --region DatabaseRegion

```

(Optional) To apply the upgrade immediately, include the `--apply-immediately` parameter in the command. You will see a response similar to the following example, and your database will become unavailable while the upgrade is being applied. For more information, see [update-relational-database](#) in the Lightsail API Reference.

```
% aws lightsail update-relational-database \
--relational-database-name "Database-Mysql-5.7" \
--relational-database-blueprint-id "mysql_8_0" \
--apply-immediately \
[--region us-east-1
{
  "operations": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb",
      "resourceName": "Database-Mysql-5.7",
      "resourceType": "RelationalDatabase",
      "createdAt": 2024-01-01T00:00:00.000000+00:00",
      "location": {
        "availabilityZone": "us-east-1a",
        "regionName": "us-east-1"
      },
      "isTerminal": true,
      "operationDetails": "",
      "operationType": "UpdateRelationalDatabase",
      "status": "Succeeded",
      "statusChangedAt": 2024-01-01T00:00:00.000000+00:00",
    }
  ]
}
```

- Enter the following command to verify that the major version upgrade is scheduled for the next database maintenance window. In the command, replace *DatabaseName* with the name of your database, and *DatabaseRegion* with the AWS Region that your database is in.

```
aws lightsail get-relational-database \
--relational-database-name DatabaseName \
--region DatabaseRegion
```

In the `get-relational-database` response, the database [state](#) informs you of a pending major version upgrade during the next maintenance window. You can locate the date and time of the next maintenance window in the [preferredMaintenanceWindow](#) section of the response.

Database instance state

```
"state": "upgrading",
"backupRetentionEnabled": true,
```

```
"pendingModifiedValues": {  
  "engineVersion": "8.0.36"
```

Maintenance window

```
"preferredMaintenanceWindow": "wed: 09:22-wed: 09:52"
```

Next steps

If you created a test database, you can delete it after you have verified that your application will work with the upgraded database. Keep the snapshot that you created of your previous database in case you need to go back to it. You should also create a snapshot of your upgraded database so that you have a new point-in-time copy of it.

Load balancers in Amazon Lightsail

A Lightsail load balancer distributes incoming web traffic among multiple Lightsail instances, in multiple Availability Zones. Load balancing increases the availability and fault tolerance of the application on your instances. You can add and remove instances from your Lightsail load balancer as your needs change, without disrupting the overall flow of requests to your application.

With Lightsail load balancing, we create a DNS host name and route any requests sent to this host name to a pool of target Lightsail instances. You can add as many target instances to your load balancer as you like, as long as you stay within your Lightsail account quotas for total number of instances.

Load balancer features

Lightsail load balancers offer the following features:

- **HTTPS encryption** — By default, Lightsail load balancers handle unencrypted (HTTP) traffic requests through port 80. Activate HTTPS encryption by attaching a validated Lightsail SSL/TLS certificate to your load balancer. This allows your load balancer to handle encrypted (HTTPS) traffic requests through port 443. For more information, see [SSL/TLS certificates](#).

The following features are available after you activate HTTPS encryption on your load balancer:

- **HTTP to HTTPS redirection** — Activate HTTP to HTTPS redirection to automatically redirect HTTP requests to an HTTPS encrypted connection. For more information, see [Configure HTTP to HTTPS redirection for your load balancer](#).
- **TLS security policies** — Configure a TLS security policy on your load balancer. For more information, see [Configuring TLS security policies on your Amazon Lightsail load balancers](#).
- **Health checking** — By default, health checks are performed on the attached instances at the root of the web application that is running on them. The health checks monitor the health of the instances so that the load balancer can send requests only to the healthy instances. For more information, see [Health checking for a Lightsail load balancer](#).
- **Session persistence** — Configure session persistence if you're storing session information locally in your website visitors' browsers. For example, you might be running a Magento e-commerce application with a shopping cart on your load-balanced Lightsail instances. If your website visitors add items to their shopping carts, and then end their sessions, when they come

back, the shopping cart items will still be there if you configured session persistence. For more information, see [Enable session persistence for a load balancer](#).

When to use load balancers

You should use a load balancer when you have a website that has occasional spikes in traffic or hosts content that can create a lot of load on an instance when many visitors are using it at once. For example, if you have an image-heavy website, you can load balance the image requests with the other page requests. That way, your pages load faster and your users are happier.

You can use a load balancer to create a highly available website. *High availability* refers to how long your website or application stays up over a given time period. If you have ever experienced a site outage, then a load balancer might help you have more uptime. You can use a Lightsail load balancer to make your application highly available by adding target instances that are distributed across multiple Availability Zones.

Fault tolerance is a related concept. If your site continues to function even after one of your instances or your database fails, it is considered tolerant. A load balancer can help you create a fault tolerant application or website.

Recommended applications for load balancing

Not all Lightsail applications need load balancers. If you decide to create a load-balanced application, you must configure your application first. For example, to prepare a LAMP stack application for load balancing, you should first create a centralized, dedicated database for all the target instances to read from and write to. You might also consider creating centralized media storage, such as a Lightsail object storage bucket. For more information, see [Configure an instance for load balancing](#).

Get started using load balancers

You can [create a load balancer](#) using the Lightsail console, the AWS Command Line Interface (AWS CLI), or the Lightsail API. You must also [configure your instances for load balancing](#).

After you create your load balancer and attach your configured instances, you can enable HTTPS using the following topic. For more information, see [Create an SSL/TLS certificate for your load balancer](#).

Create a Lightsail load balancer and attach instances to it

Create a load balancer to add redundancy to your application or to handle more web traffic. After the load balancer is created, you can attach the Lightsail instances that you want to balance. To learn more, see [Load balancers](#)

Prerequisites

Before you begin, make sure you've prepared your Lightsail instances for load balancing. For more information, see [Configure an instances for load balancing](#).

Create a load balancer

1. Sign in to the [Lightsail console](#).
2. Choose the **Networking** tab.
3. Choose **Create load balancer**.
4. Confirm the AWS Region where the load balancer will be created, or choose **Change region** to select a different region.

Note

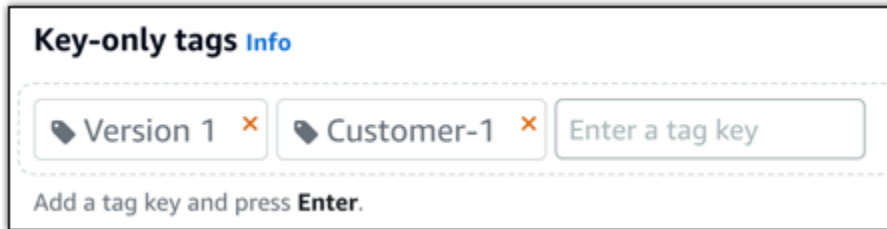
By default, the load balancer will be created with port 80 open to accept HTTP requests. After the load balancer is created, you can create an SSL/TLS certificate and configure HTTPS. For more information, see [Create an SSL/TLS certificate for your load balancer](#)

5. Enter a name for your load balancer.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
6. Choose one of the following options to add tags to your load balancer:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



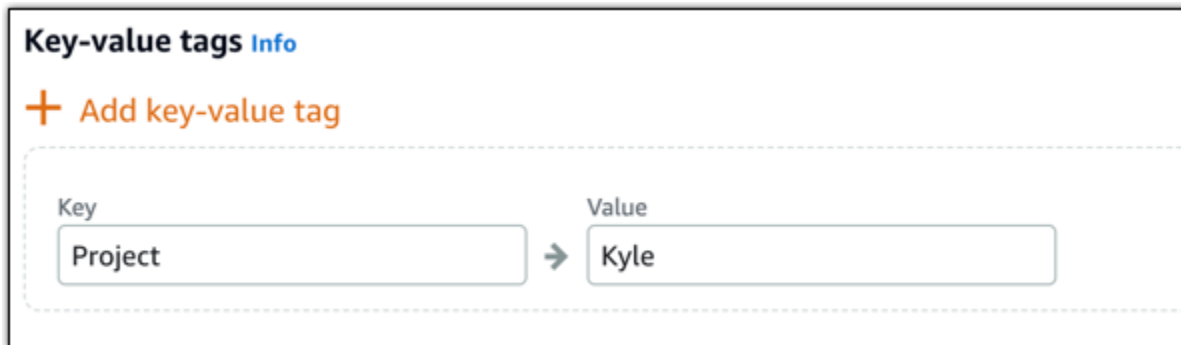
Key-only tags Info

Version 1 × Customer-1 × Enter a tag key

Add a tag key and press **Enter**.

- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key Value

Project → Kyle

Note

For more information about key-only and key-value tags, see [Tags](#).

7. Choose **Create load balancer**.

Attach an instance to your load balancer

After your load balancer is created, Lightsail takes you to the load balancer management page. If you need to find that page again, choose the **Networking** tab on the Lightsail home page, and then choose the name of your Lightsail load balancer to manage it.

Note

Your Lightsail instance must be running before you can successfully attach it to your load balancer.

1. On the load balancer management page, choose **Target instances**.
2. Choose an instance in the **Target instances** drop-down menu.
3. Choose **Attach**. Attachment can take several minutes.

Attach another instance to the load balancer by choosing **Attach another**, and then repeating the preceding steps.

Next steps

After the load balancer is created, and your instances are attached, complete the following next steps to configure your load balancer:

- [Create an SSL/TLS certificate for your load balancer](#)
- [Customize health checks for your load balancer](#)

If you experience issues with your load balancer, see [Troubleshoot your load balancer](#)

Create an SSL/TLS certificate for your Amazon Lightsail load balancer

After you create a Lightsail load balancer, you can attach a Transport Layer Security (TLS) certificate to enable HTTPS. The SSL/TLS certificate lets your load balancer handle encrypted web traffic so that you can provide a more secure experience for your users. To learn more, see [SSL/TLS certificates](#).

Prerequisites

Before you get started, you will need the following.

- A Lightsail load balancer. To learn more, see [Create a load balancer](#).

Create the certificate request

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose **Networking**.
3. Choose the name of the load balancer for which you want to configure an SSL/TLS certificate.
4. Choose the **Custom domains** tab.
5. Choose **Create certificate**.
6. Enter a name for your certificate or accept the default.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
7. Enter your primary domain (`www.example.com`), and up to 9 alternate domains or subdomains.

For more information, see [Add alternate domains and subdomains to your SSL/TLS certificate](#)

8. Choose **Create certificate**.

Lightsail begins the validation process. You have 72 hours to verify that you own your domain.

After you create your certificate, you see the certificate along with the domain name and all your alternate domains and subdomains. You need to create a DNS record for each domain and subdomain.

Next step

- [Verify that you own your domain](#)

Topics

- [Add alternate domains and subdomains to your SSL/TLS certificate in Lightsail](#)
- [Verify an SSL/TLS certificate in Amazon Lightsail](#)
- [Attach a validated SSL/TLS certificate to your Amazon Lightsail load balancer](#)

- [Delete an SSL/TLS certificate in Amazon Lightsail](#)

Add alternate domains and subdomains to your SSL/TLS certificate in Lightsail

When you create your SSL/TLS certificate for your Lightsail load balancer, you can add alternate domains and subdomains to it. These alternate names help ensure that all traffic to your load balancer is encrypted.

When you specify a primary domain, you can use a fully qualified domain name such as `www.example.com` or an apex domain name such as `example.com`.

The total number of domains and subdomains must not exceed 10, so you can add up to 9 alternate domains and subdomains to your certificate. You might want to add entries similar to the following list.

- `example.com`
- `example.net`
- `blog.example.com`
- `myexamples.com`

To create a certificate with alternate domains and subdomains

1. If you don't have one yet, [Create a load balancer](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose your Lightsail load balancer.
4. Choose the **Custom domains** tab.
5. Choose **Create certificate**.
6. Enter a name for your certificate or accept the default name.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.

- Can include alphanumeric characters, numbers, periods, dashes, and underscores.
7. Enter your primary domain (`www.example.com`), and up to 9 alternate domains or subdomains.
 8. Choose **Create certificate**.

Once created, you have 72 hours to verify that you own your domain.

Next steps

- [Verify domain ownership using DNS](#)

Once verified, you can select your validated certificate to associate it with your Lightsail load balancer.

- [Enable session persistence](#)

Verify an SSL/TLS certificate in Amazon Lightsail

After you create an SSL/TLS certificate in Lightsail, you need to verify that you control all the domains and subdomains that you added to the certificate.

Contents

- [Step 1: Create a Lightsail DNS zone for your domain](#)
- [Step 2: Add records to your domain's DNS zone](#)
- [Next step](#)

Step 1: Create a Lightsail DNS zone for your domain

If you haven't done so already, create a Lightsail DNS zone for your domain. For more information, see [Create a DNS zone to manage your domain's DNS records](#)

Step 2: Add records to your domain's DNS zone

The certificate that you created provides a set of canonical name (CNAME) records. You add these records to your domain's DNS zone to verify that you own or control that domain.

⚠ Important

Lightsail will attempt to automatically verify that you control the domains or subdomains you specified while creating the certificate. After you select **Create certificate**, the CNAME records will be added to your domain's DNS zone. The certificate's status will change from **Attempting to validate your certificate**, to **Valid, in use** if automatic validation is successful.

Proceed to the following steps if automatic validation fails.

In the following steps, we'll show you how to get the CNAME records and add them to your domain's DNS zone in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose **Account** on the top navigation menu.
3. Choose **Account** in the dropdown menu.
4. Choose the **Certificates** tab.
5. Find the certificate that you want to verify, and make note of the **Name** and **Value** of the CNAME records that you must add for each domain

Press **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using Mac, to copy them to your clipboard.

example.com
SSL certificate, example.com
Requested on: January 15, 2019, 2:57 PM

Status:  **Validation in progress...**

You must prove you control the domains and subdomains specified in this certificate before it can be used for HTTPS encryption.

Please create a DNS record for each domain with the following values:

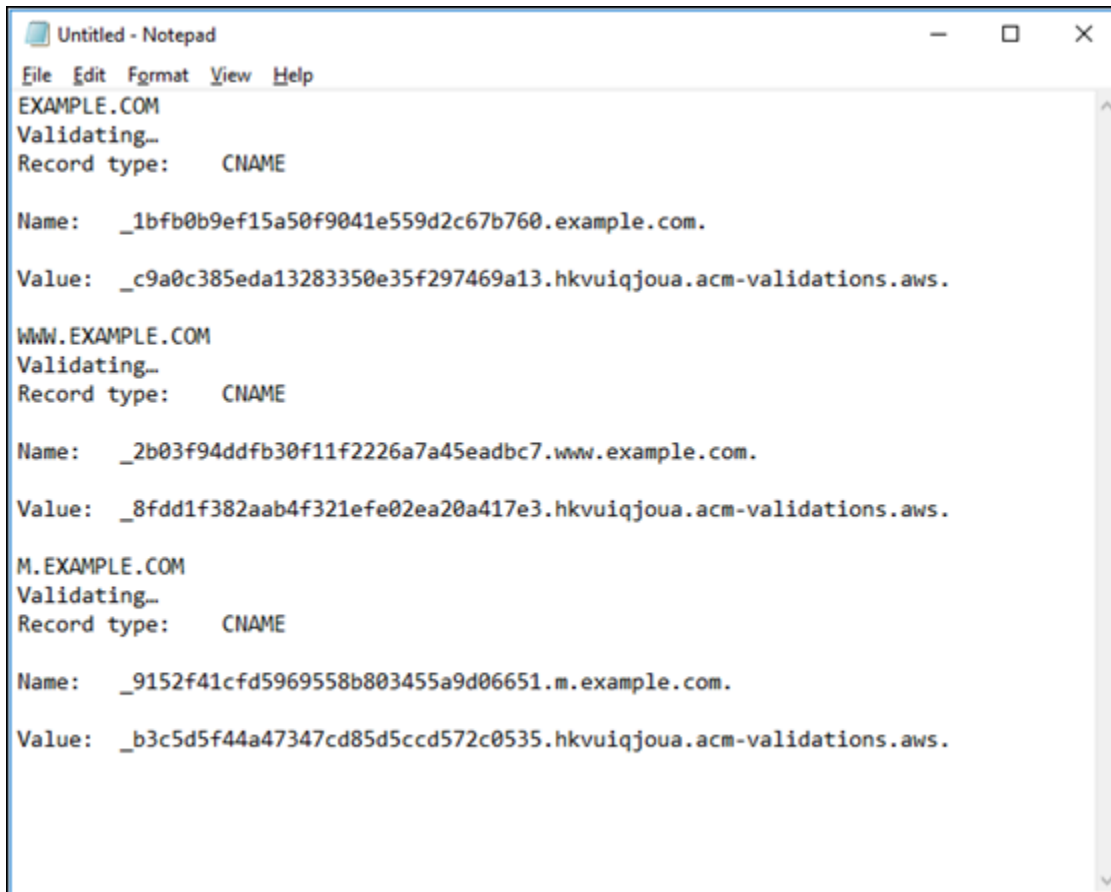
EXAMPLE.COM Validating...
Record type: CNAME
Name: `_1bfb0b9ef15a50f9041e559d2c67b760.example.com.`
Value: `c9a0c385eda13283350e35f297469a13.hkvuiqjoua.acm-validations.aws.`

WWW.EXAMPLE.COM Validating...
Record type: CNAME
Name: `_2b03f94ddf30f11f2226a7a45eadbc7.www.example.com.`
Value: `_8fdd1f382aab4f321efe02ea20a417e3.hkvuiqjoua.acm-validations.aws.`

M.EXAMPLE.COM Validating...
Record type: CNAME
Name: `_9152f41cfd5969558b803455a9d06651.m.example.com.`
Value: `_b3c5d5f44a47347cd85d5ccd572c0535.hkvuiqjoua.acm-validations.aws.`

6. Open a text editor, such as Notepad if you're using Windows, or TextEdit if you're using Mac. In the text file, press **Ctrl+V** if you're using Windows, or **Cmd+V** if you're using Mac, to paste the values into the text file.

Leave this text file open; you will need these CNAME values when adding the records to your domain's DNS zone later in this guide.



```
Untitled - Notepad
File Edit Format View Help
EXAMPLE.COM
Validating...
Record type: CNAME

Name: _1bfb0b9ef15a50f9041e559d2c67b760.example.com.
Value: _c9a0c385eda13283350e35f297469a13.hkvuijqjou.acm-validations.aws.

WWW.EXAMPLE.COM
Validating...
Record type: CNAME

Name: _2b03f94ddfb30f11f2226a7a45eadbc7.www.example.com.
Value: _8fdd1f382aab4f321efe02ea20a417e3.hkvuijqjou.acm-validations.aws.

M.EXAMPLE.COM
Validating...
Record type: CNAME

Name: _9152f41cfd5969558b803455a9d06651.m.example.com.
Value: _b3c5d5f44a47347cd85d5ccd572c0535.hkvuijqjou.acm-validations.aws.
```

7. Choose **Home** on the top navigation bar of the Lightsail console.
8. Choose **Domains & DNS** on the Lightsail home page.
9. Choose the DNS zone for the domain that will use the certificate.
10. Choose **Add record** in the **DNS records** tab.
11. Choose **CNAME** for the record type.
12. Toggle to the text file that contains the CNAME records for your certificates.

Copy the **Name** of the CNAME record. For example, `_1bfb0b9ef15a50f9041e559d2c67b760`.

13. Toggle to the DNS records page and paste the **Name** into the **Record name** field.

Important

Adding a CNAME record that contains the domain name (such as `.example.com`) will result in duplication of the domain name (such as `.example.com.example.com`). To

avoid duplication, edit the entry so that only the part of the CNAME that you need is added. This would be `_1bfb0b9ef15a50f9041e559d2c67b760`.

14. Copy the **Value** of the CNAME record. For example, `_c9a0c385eda13283350e35f297469a13.hkvuiqjoua.acm-validations.aws..`
15. Toggle to the DNS records page and paste the **Value** into the **Route traffic to** field.
16. Choose **Save** to add the record.
17. If you have alternate subdomains, choose **Add record** to add another record.

Note



To learn more about alternate domains or subdomains, see [Add alternate domains and subdomains to your SSL/TLS certificate in Amazon Lightsail](#).

18. Repeat steps 11 - 17 to add the CNAME record(s) for the alternate subdomains.


You can also [add an alias \(A\) record to point to your load balancer](#), or other Lightsail resources while you're on the DNS zone management page.



When finished, your DNS zone should look like the following screenshot.

[+ Add record](#)

A record  



Associate your domain or a subdomain with an IP address.

Subdomain: @.example.com Resolves to:  LoadBalancer-Oregon-1


CNAME record  



Create a subdomain alias of example.com and point it to another domain.

Subdomain: _dead6a124... .example.com Maps to: _be133b0a0899fb7b6bf79d9741d...

A record  

Associate your domain or a subdomain with an IP address.


Subdomain: www.example.com Resolves to:  LoadBalancer-Oregon-1

CNAME record  



Create a subdomain alias of example.com and point it to another domain.

Subdomain: _bb150425... .example.com Maps to: _9317035fb90049adff91310d7a1...

After some time, your domain is verified and you will see the following message on the certificate.

Certificates 

You may create and store up to two SSL/TLS certificates per load balancer to choose from

 **example.com** 

SSL certificate, example.com
Requested on: January 14, 2019, 3:13 PM

Status: **Valid, in use**

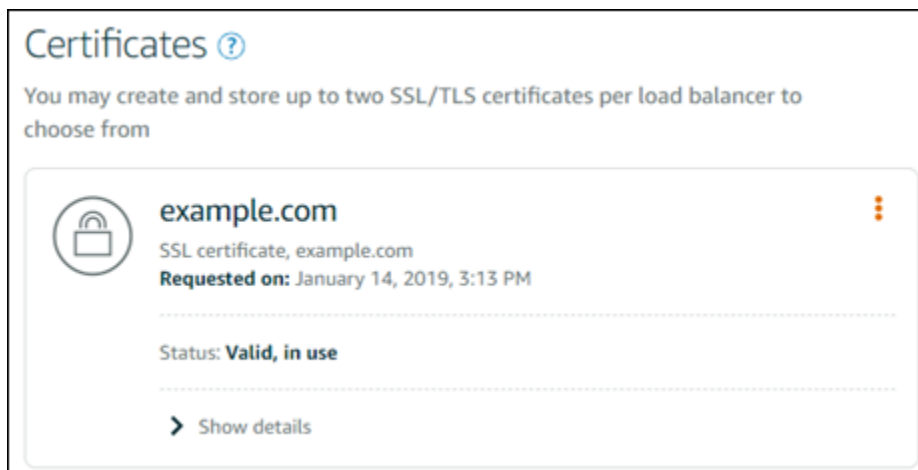
[> Show details](#)

Next step

Once your domain is verified, you are ready to [Attach a validated SSL/TLS certificate to your load balancer](#).

Attach a validated SSL/TLS certificate to your Amazon Lightsail load balancer

After you verify that you control your domain, the certificate's status will change to **Valid**.



Your next step is to attach the certificate to your Lightsail load balancer.

1. From the Lightsail home page, choose **Networking**.
2. Choose your load balancer.
3. Choose the **Custom domains** tab.
4. In the **Certificates** section, choose **Attach certificate**.
5. Select a certificate from the dropdown list.
6. Choose **Attach** to attach the certificate.

Delete an SSL/TLS certificate in Amazon Lightsail

You can delete an SSL/TLS certificate that you're no longer using. For example, your certificate might be expired and you've already attached an updated certificate that's validated. If you want to duplicate your certificate before deleting it, you can choose **Duplicate** from the same shortcut menu in step 5, below.

Important

If the certificate you're deleting is valid and in use, your load balancer will no longer be able to handle encrypted (HTTPS) traffic. Your Lightsail load balancer will still support unencrypted (HTTP) traffic.

Deleting an SSL/TLS certificate is final and can't be undone. You have a quota of certificates you can create over a 365-day period. For more information, see [Quotas](#) in the AWS Certificate Manager User Guide.

1. On the Lightsail home page, choose **Networking**.
2. Choose the load balancer where your SSL/TLS certificate is attached.
3. Choose the **Inbound traffic** tab on your load balancer's management page.
4. In the **Certificates** section of the page, choose the ellipsis icon (:) for the certificate that you want to delete, and choose **Delete**.

The **Delete** option is unavailable if the certificate you want to delete is in use. To delete certificates that are in use, you need to first change the certificate of the load balancer that is using the certificate, or disable HTTPS on the load balancer that is using the certificate.

Update Amazon Lightsail load balancer settings

When you create a Lightsail load balancer, you choose the AWS Region and the name. This topic instructs you how to update your load balancer to enable more options.

If you haven't done so already, you'll need to create a load balancer. [Create a load balancer](#)

Health checks

The first thing you're going to want to do is [Configure an instance for load balancing](#). Once that's done, you can attach an instance to your load balancer. Attaching an instance starts the health checking process, and you get a **Passed** or **Failed** message on the load balancer management page.

Target Instances Inbound Traffic Delete

Target Instances

Traffic will be evenly distributed to the following instances:

Attach another

example-1 Detach
8 GB RAM, 2 vCPUs, 80 GB SSD
WordPress

Health Check: **Passed**

example-2 Detach
8 GB RAM, 2 vCPUs, 80 GB SSD
WordPress

Health Check: **Passed**

Your instances will receive traffic from this load balancer on port 80
[Learn more about load balancing](#)

You can also customize your health check path. For example, if your home page loads slowly or has a lot of images on it, you can configure Lightsail to check a different page that loads faster.

[Customize load balancer health check paths](#)

Encrypted traffic (HTTPS)

You can set up HTTPS to create a more secure experience for your website users. It's a three-step process to create and validate an SSL/TLS certificate once you set up your load balancer.

[Learn more about HTTPS](#)

Session persistence

Session persistence is useful if you're storing session information locally in the user's browser. For example, you might be running a Magento e-commerce application with a shopping cart on Lightsail. If you turn on session persistence, your users can add items to their shopping carts, end their sessions, and still find the items in their carts when they come back.

You can also adjust the cookie duration for the persistent session. This is useful if you want to have a particularly long or short duration. For more information, see [Enable session persistence for a load balancer](#).

Configure a Lightsail instance for load balancing

Before you attach instances to your Lightsail load balancer, you need to evaluate your application's configuration. For example, load balancers often work better when the data tier is separated from the rest of the application. This topic tells you about each Lightsail instance and makes recommendations about whether to load balance (or *horizontally scale*) and how to best configure your application.

General guidelines: Applications that use a database

For Lightsail applications that use a database, we recommend that you separate the database instance from the rest of your application, so that you only have one database instance. The main reason is that you want to avoid writing data to more than one database. If you don't create a single database instance, then the data will be written to the database on whichever instance the user happens to hit.

WordPress

Horizontally scale? Yes, for either a WordPress blog or website.

Configuration recommendations before using a Lightsail load balancer

- Separate your database so that every WordPress instance running behind the load balancer is storing and retrieving information from the same place. If you need more performance from your database, you can replicate or change the processing power or memory independently of your web server.
- Offload your files and static content to a Lightsail bucket. To do this, you must install the WP Offload Media Lite plugin on your WordPress website and configure it to connect to your Lightsail bucket. For more information, see [Tutorial: Connect a WordPress instance to a storage bucket](#).

Node.js

Horizontally scale? Yes, with some considerations.

Configuration recommendations before using a Lightsail load balancer

- In Lightsail, the Node.js stack packaged by Bitnami contains Node.js, Apache, Redis (an in-memory database), and Python. Depending on the application you're deploying, you can load balance across a few servers. However, you would need to configure a load balancer to balance the traffic among all the web servers and move Redis to another server.
- Split the Redis server to another server to communicate with all the instances. Add a database server, if necessary.
- One of the primary use cases for Redis is to cache data locally so you don't have to constantly hit the central database. We recommend that you enable session persistence to leverage the performance improvement from Redis. For more information, see [Enable session persistence for a load balancer](#).
- You can also have a shared Redis node, so you can also share a node or use a local cache on each machine using session persistence.
- Consider including the `mod_proxy_balancer` in the Apache server, if you want to deploy a load balancer using Apache.

For more information, see [Scaling Node.js applications](#).

Magento

Horizontally scale? Yes.

Configuration recommendations before using a Lightsail load balancer

- You can use an AWS reference deployment of Magento that uses additional components, such as an Amazon RDS database: [Terraform Magento Adobe Commerce on AWS](#).
- Be sure to enable session persistence. Magento uses a shopping cart, and this helps ensure that customers who make multiple visits across more than one session will retain items in their shopping carts when they return for a new session. For more information, see [Enable session persistence for a load balancer](#).

GitLab

Horizontally scale? Yes, with considerations.

Configuration recommendations before using a Lightsail load balancer

You must have the following:

- A Redis node running and ready to use
- A shared network storage server (NFS)
- A centralized database (MySQL or PostgreSQL) for the application. See the general guidelines about databases, above.

For more information, see [High Availability](#) on the *GitLab* website.

Note

The shared network storage server (NFS) referred to above, is not currently available with the GitLab blueprint.

Drupal

Horizontally scale? Yes. Drupal has an official document describing how to horizontally scale your application: [Server Scaling](#).

Configuration recommendations before using a Lightsail load balancer

You must set up a Drupal module to synchronize files among different instances. The Drupal website features several modules, but they may be more suitable for prototyping as opposed to production use.

Use a module that lets you store your files in Amazon S3. This gives you a centralized place for your files, rather than keeping separate copies on each target instance. That way, if you edit your files, the updates get picked up from the centralized store and your users see the same files, regardless of which instance they hit.

- [Amazon S3 File System](#)
- [Content Synchronization](#)

For more information, see [Scaling Drupal horizontally and in cloud](#).

LAMP stack

Horizontally scale? Yes.

Configuration recommendations before using a Lightsail load balancer

- You should create a database on a separate instance. All the instances behind the load balancer should point to this separate database instance so they store and retrieve information from the same place.
- Depending on the application that you want to deploy, think about how to share the file system (NFS, Lightsail block storage disks, or Amazon S3 storage).

MEAN stack

Horizontally scale? Yes.

Configuration recommendations before using a Lightsail load balancer

Move MongoDB to another machine and configure a mechanism to share the root document among the Lightsail instances.

Redmine

Horizontally scale? Yes.

Configuration recommendations before using a Lightsail load balancer

- Get the [Redmine_S3 plugin](#) to store the attachments on Amazon S3 instead of on the local file system.
- Separate the database to a different instance.

Nginx

Horizontally scale? Yes.

You can have one or more Lightsail instances running Nginx and attached to a Lightsail load balancer. For more information, see [Scaling Web Applications with NGINX, Part 1: Load Balancing](#).

Joomla!

Horizontally scale? Yes, with considerations.

Configuration recommendations before using a Lightsail load balancer

Although there is no official documentation on the Joomla website, there are some discussions on their community forums. Some users managed to horizontally scale their Joomla instances having a cluster with the following configuration:

- A Lightsail load balancer configured to enable session persistence. For more information, see [Enable session persistence for a load balancer](#).
- Several Lightsail instances running Joomla attached to the load balancer with the document root of Joomla! synchronized. You can do this using tools like Rsync, having an NFS server that is in charge of synchronizing the content among all Lightsail instances, or sharing files using AWS.
- Several database servers configured with a replication cluster.
- The same cache system configured in each Lightsail instance. There are some useful extensions, such as [JotCache](#).

Configure TLS security policies on your Amazon Lightsail load balancer

After you enable HTTPS on your Amazon Lightsail load balancer, you can configure a TLS security policy for the encrypted connections. This guide provides information about the security policies that you can configure on Lightsail load balancers, and the procedures for updating your load balancer's security policy. For more information about load balancers, see [Load balancers](#).

Security policies overview

Lightsail load balancing uses a Secure Socket Layer (SSL) negotiation configuration, known as a security policy, to negotiate SSL connections between a client and the load balancer. A security policy is a combination of protocols and ciphers. The protocol establishes a secure connection between a client and a server and ensures that all data passed between the client and your load balancer is private. A cipher is an encryption algorithm that uses encryption keys to create a coded message. Protocols use several ciphers to encrypt data over the internet. During the connection negotiation process, the client and the load balancer present a list of ciphers and protocols that they each support, in order of preference. By default, the first cipher on the server's list that matches any one of the client's ciphers is selected for the secure connection. Lightsail load balancers do not support SSL renegotiation for client or target connections.

The TLS-2016-08 security policy is configured by default when you enable HTTPS on a Lightsail load balancer. You can configure a different security policy as needed, as described later in this

guide. You can choose the security policy that is used for only for front-end connections. The TLS-2016-08 security policy is always used for backend connections. Lightsail load balancers do not support custom security policies.

Supported security policies and protocols

Lightsail load balancers can be configured with the following security policies and protocols:

Security policies	TLS-2016-08 (default)	TLS-FS-1-2-Res-2019-08
TLS Protocols		
Protocol-TLSv1	✓	
Protocol-TLSv1.1	✓	
Protocol-TLSv1.2	✓	✓
TLS Ciphers		
ECDHE-ECDSA-AES128-GCM-SHA256	✓	✓
ECDHE-RSA-AES128-GCM-SHA256	✓	✓
ECDHE-ECDSA-AES128-SHA256	✓	✓
ECDHE-RSA-AES128-SHA256	✓	✓
ECDHE-ECDSA-AES128-SHA	✓	
ECDHE-RSA-AES128-SHA	✓	
ECDHE-ECDSA-AES256-GCM-SHA384	✓	✓
ECDHE-RSA-AES256-GCM-SHA384	✓	✓
ECDHE-ECDSA-AES256-SHA384	✓	✓
ECDHE-RSA-AES256-SHA384	✓	✓
ECDHE-RSA-AES256-SHA	✓	
ECDHE-ECDSA-AES256-SHA	✓	
AES128-GCM-SHA256	✓	
AES128-SHA256	✓	
AES128-SHA	✓	

Complete the prerequisites

Complete the following prerequisites if you haven't already:

- Create a load balancer and attach instances to it. For more information, see [Create a load balancer and attach instances to it](#).
- Create an SSL/TLS certificate and attach it to your load balancer to enable HTTPS. For more information, see [Create an SSL/TLS certificate for your Lightsail load balancer](#). For more information about certificates, see [SSL/TLS certificates](#).

Configure a security policy using the Lightsail console

Complete the following procedure to configure a security policy using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the load balancer for which you want to configure a TLS security policy.
4. Choose the **Inbound traffic** tab.
5. Choose **Change protocols** under the **TLS security protocols** section of the page.
6. Select one of the following options in the **Supported protocols** dropdown menu:
 - **TLS version 1.2** — This option is the most secure but older browsers might be unable to connect.
 - **TLS version 1.0, 1.1, and 1.2** — This option offers the most compatibility with browsers.
7. Choose **Save** to apply the selected protocol to your load balancer.

Your change takes a few moments to become effective.

Configure a security policy using the AWS CLI

Complete the following procedure to configure a security policy using the AWS Command Line Interface (AWS CLI). You do this by using the `update-load-balancer-attribute#command`. For more information, see [update-load-balancer-attribute#](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to change the TLS security policy for your load balancer.

```
aws lightsail update-load-balancer-attribute --load-balancer-name LoadBalancerName
--attribute-name TlsPolicyName --attribute-value AttributeValue
```

In the command, replace the following example text with your own:

- *LoadBalancerName* with the name of the load balancer for which you want to change the TLS security policy.
- *AttributeValue* with the TLS-2016-08 or TLS-FS-1-2-Res-2019-08 security policy.

Note

The TlsPolicyName attribute in the command specifies that you wish to edit the TLS security policy that is configured on the load balancer.

Example:

```
aws lightsail update-load-balancer-attribute --load-balancer-name MyLoadBalancer --
attribute-name TlsPolicyName --attribute-value TLS-2016-08
```

Your change takes a few moments to become effective.

Configure HTTP to HTTPS redirect for a Lightsail load balancer

After you configure HTTPS on your Amazon Lightsail load balancer, you can configure an HTTP to HTTPS redirect so that users who browse to your website or web application using an HTTP connection are automatically redirected to the encrypted HTTPS connection. For more information about load balancers, see [Load balancers](#).

Complete the prerequisites

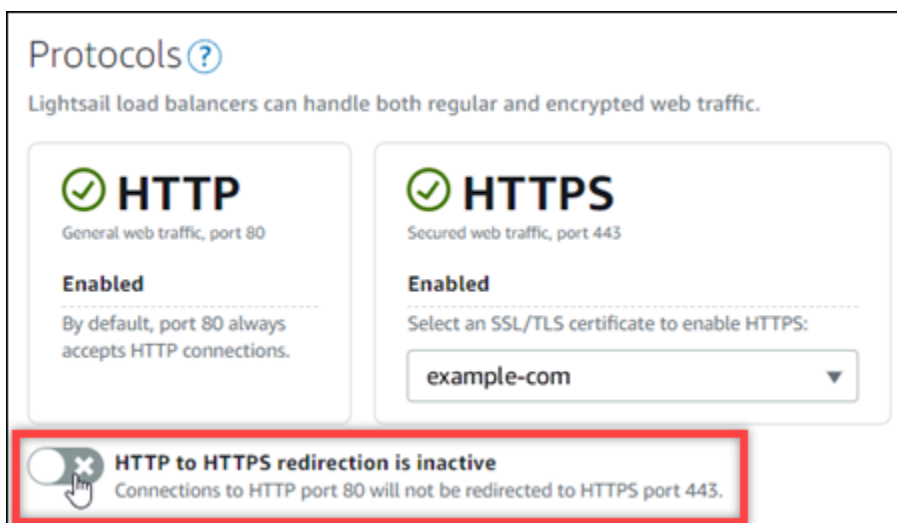
Complete the following prerequisites if you haven't already:

- Create a load balancer and attach instances to it. For more information, see [Create a load balancer and attach instances to it](#).
- Create an SSL/TLS certificate and attach it to your load balancer to enable HTTPS. For more information, see [Create an SSL/TLS certificate for your Lightsail load balancer](#). For more information about certificates, see [SSL/TLS certificates](#).

Configure HTTPS redirection on your load balancer using the Lightsail console

Complete the following procedure to configure HTTPS redirection on your load balancer using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the load balancer for which you want to configure HTTPS redirection.
4. Choose the **Inbound traffic** tab.
5. In the **Protocols** section of the page, you can perform one of the following actions:



- Toggle the direction option to active to turn on HTTP to HTTPS redirection.
- Toggle the direction option to inactive to turn off HTTP to HTTPS redirection.

Your change takes a few moments to become effective.

Configure HTTP to HTTPS redirect for a load balancer with the AWS CLI

Complete the following procedure to configure HTTPS redirection on your load balancer using the AWS Command Line Interface (AWS CLI). You do this by using the `update-load-balancer-attribute` command. For more information, see [update-load-balancer-attribute](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to configure HTTPS redirection on your load balancer.

```
aws lightsail update-load-balancer-attribute --load-balancer-name LoadBalancerName
--attribute-name HttpsRedirectionEnabled --attribute-value AttributeValue
```

In the command, replace the following example text with your own:

- *LoadBalancerName* with the name of the load balancer for which you want to activate or deactivate HTTP to HTTPS redirection.
- *AttributeValue* with `true` to activate redirection, or `false` to deactivate redirection.

Note

The `HttpsRedirectionEnabled` attribute in the command specifies that you wish to edit whether HTTPS redirection is enabled or disabled for the specified load balancer.

Examples:

- To activate HTTP to HTTPS redirection on your load balancer:

```
aws lightsail update-load-balancer-attribute --load-balancer-name MyLoadBalancer
--attribute-name HttpsRedirectionEnabled --attribute-value true
```

- To deactivate HTTP to HTTPS redirection on your load balancer:

```
aws lightsail update-load-balancer-attribute --load-balancer-name MyLoadBalancer
--attribute-name HttpsRedirectionEnabled --attribute-value false
```

Your change takes a few moments to become effective.

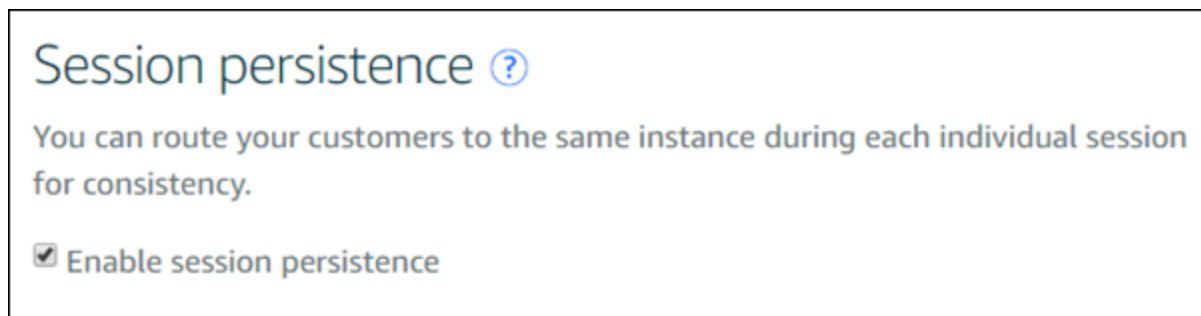
Enable session persistence for a Lightsail load balancer

You can enable *session persistence* for your users. This is helpful if you're storing session information locally in the user's browser. For example, you might be running a Magento e-commerce application with a shopping cart on Lightsail. If you turn on session persistence, your users can add items to their shopping carts, leave the site, and still find the items in their carts when they come back.

You can also adjust the cookie duration using the AWS Command Line Interface (AWS CLI) or the Lightsail API.

Enable session persistence

1. On the Lightsail home page, choose **Networking**.
2. Choose your load balancer to manage it.
3. Choose the **Inbound traffic** tab.
4. Choose **Enable session persistence**.



Adjust the cookie duration

You can also adjust the cookie duration for the persistent session. This is useful if you want to have a particularly long or short duration. For example, for many ecommerce sites the duration is quite long. This lets customers leave and come back without losing items in their shopping carts.

If you haven't done so already, set up the AWS CLI and configure it.

[Configure the AWS Command Line Interface to work with Amazon Lightsail](#)

1. Open a command prompt or a terminal window.
2. Type the following AWS CLI command to increase the cookie duration to three days (259,200 seconds).

```
aws lightsail update-load-balancer-attribute --load-balancer-name LoadBalancerName
--attribute-name SessionStickiness_LB_CookieDurationSeconds --attribute-value
259200
```

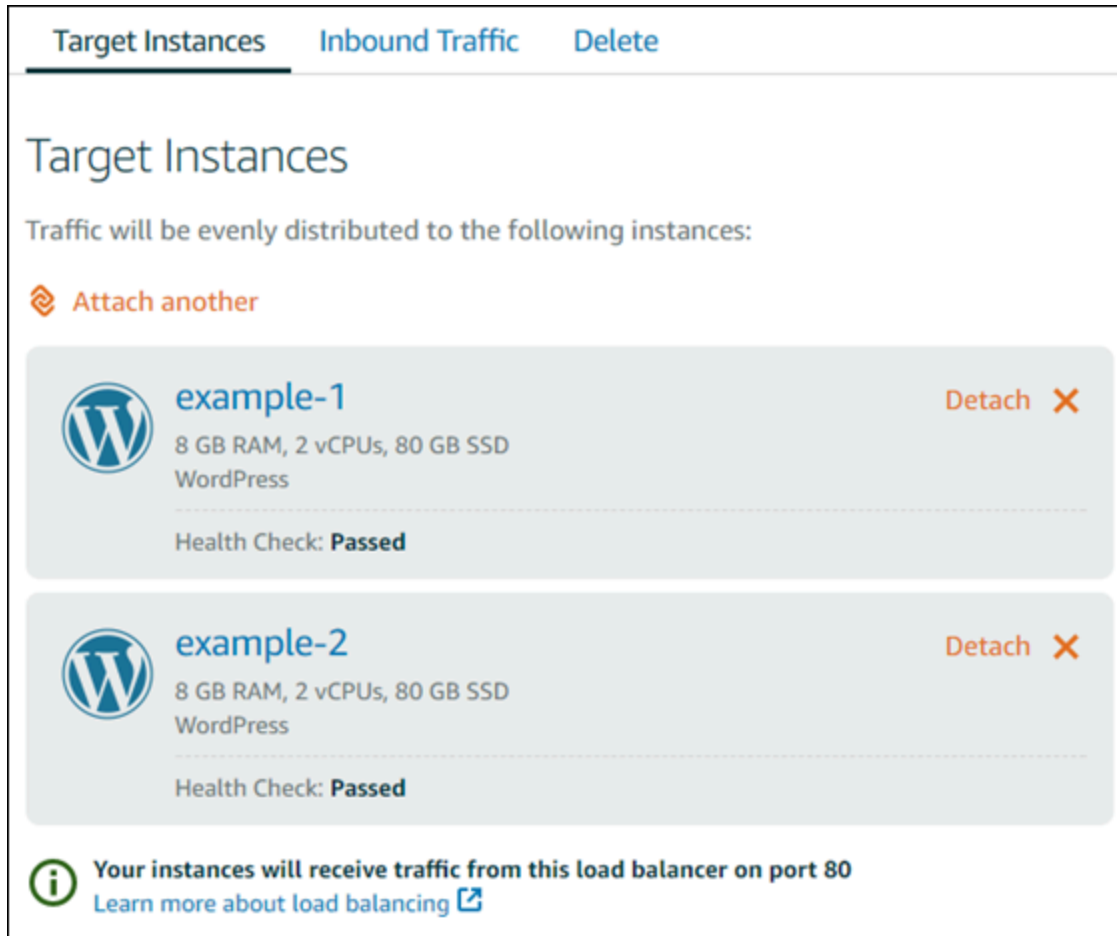
In the command, replace *LoadBalancerName* with the name of your load balancer.

If successful, you should see the following response.

```
{
  "operations": [
    {
      "status": "Succeeded",
      "resourceType": "LoadBalancer",
      "isTerminal": true,
      "operationDetails": "SessionStickiness_LB_CookieDurationSeconds",
      "statusChangedAt": 1511758936.174,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "operationType": "UpdateLoadBalancerAttribute",
      "resourceName": "example-load-balancer",
      "id": "681c2bd9-9a51-402b-8ad2-12345EXAMPLE",
      "createdAt": 1511758936.174
    }
  ]
}
```

Amazon Lightsail load balancer health checks

Health checking starts as soon as you attach your Lightsail instances to your load balancer, and it occurs every 30 seconds thereafter. You can see the health check status on the load balancer management page.



The screenshot displays the 'Target Instances' tab of a load balancer management page. At the top, there are three tabs: 'Target Instances' (selected), 'Inbound Traffic', and 'Delete'. Below the tabs, the title 'Target Instances' is followed by the text 'Traffic will be evenly distributed to the following instances:'. There is an orange icon and the text 'Attach another'. Two instance cards are shown, each with a WordPress logo, the name 'example-1' and 'example-2', specifications '8 GB RAM, 2 vCPUs, 80 GB SSD', and 'WordPress'. Below each card, it says 'Health Check: Passed'. To the right of each card is a 'Detach' button with a red 'X' icon. At the bottom, there is an information icon and the text 'Your instances will receive traffic from this load balancer on port 80' with a link 'Learn more about load balancing'.

Customize your health check path

You might want to customize your health check path. For example, if your home page loads slowly or has a lot of images on it, you can configure Lightsail to check a different page that loads faster.

1. On the Lightsail home page, choose **Networking**.
2. Choose your load balancer to manage it.
3. On the **Target instances** tab, choose **Customize health checking**.
4. Type a valid path for your health check, and then choose **Save**.



Health check metrics

The following metrics can help you diagnose health check problems. Use the AWS Command Line Interface or the Lightsail API to return information about the specific health check metric.

- **ClientTLSNegotiationErrorCount** - The number of TLS connections initiated by the client that did not establish a session with the load balancer. Possible causes include a mismatch of ciphers or protocols.

Statistics: The most useful statistic is Sum.

- **HealthyHostCount** - The number of target instances that are considered healthy.

Statistics: The most useful statistic are Average, Minimum, and Maximum.

- **UnhealthyHostCount** - The number of target instances that are considered unhealthy.

Statistics: The most useful statistic are Average, Minimum, and Maximum.

- **HTTPCode_LB_4XX_Count** - The number of HTTP 4XX client error codes that originate from the load balancer. Client errors are generated when requests are malformed or incomplete. These requests have not been received by the target instance. This count does not include any response codes generated by the target instances.

Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average all return 1.

- **HTTPCode_LB_5XX_Count** - The number of HTTP 5XX server error codes that originate from the load balancer. This count does not include any response codes generated by the target instances.

Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average all return 1. Note that Minimum, Maximum, and Average all return 1.

- **HTTPCode_Instance_2XX_Count** - The number of HTTP response codes generated by the target instances. This does not include any response codes generated by the load balancer.

Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average all return 1.

- **HTTPCode_Instance_3XX_Count** - The number of HTTP response codes generated by the target instances. This does not include any response codes generated by the load balancer.

Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average all return 1.

- **HTTPCode_Instance_4XX_Count** - The number of HTTP response codes generated by the target instances. This does not include any response codes generated by the load balancer.

Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average all return 1.

- **HTTPCode_Instance_5XX_Count** - The number of HTTP response codes generated by the target instances. This does not include any response codes generated by the load balancer.

Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average all return 1.

- **InstanceResponseTime** - The time elapsed, in seconds, after the request leaves the load balancer until a response from the target instance is received.

Statistics: The most useful statistic is Average.

- **RejectedConnectionCount** - The number of connections that were rejected because the load balancer had reached its maximum number of connections.

Statistics: The most useful statistic is Sum.

- **RequestCount** - The number of requests processed over IPv4. This count includes only the requests with a response generated by a target instance of the load balancer.

Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average all return 1.

Topics

- [Lightsail load balancer health check status](#)

Lightsail load balancer health check status

By default, Lightsail performs health checks on your instances at the root ("/") of your web application. The health checks are used to monitor the health of the registered instances so that the load balancer can send requests only to the healthy instances. The health checks start as soon as you attach the instances to your load balancer.

One of the following statuses is returned.

- Passed
- Failed

If your health check fails, you can try to figure out what is wrong by using the AWS Command Line Interface or the Lightsail API. See our troubleshooting guide for more information.

Customize your health check path

You might want to customize your health check path. For example, if your home page loads slowly or has a lot of images on it, you can configure Lightsail to check a different page that loads faster.

1. On the Lightsail home page, choose **Networking**.
2. Choose your load balancer to manage it.
3. On the **Target instances** tab, choose **Customize health checking**.
4. Type a valid path for your health check, and then choose **Save**.

Customize Health Check

Load balancers test the health of attached instances by attempting an HTTP connection to the path below. If the connection succeeds, the instance is considered healthy and the load balancer will send it traffic.

You can choose the path the load balancers use for health checking:

`http://{instance IP address}/`

[Why would I customize my health check path?](#)

Save Cancel

Detach instances from a Lightsail load balancer

If you no longer want to have an instance attached to your Lightsail load balancer, you can detach it. When you detach a Lightsail instance from a load balancer, we wait until the specified instances are no longer needed before detaching.

1. On the Lightsail home page, choose **Networking**.
2. Choose the load balancer you want to manage.
3. On the **Target instances** tab, choose **Detach** next to the load balancer you want to detach.

Delete a Lightsail load balancer

You can delete a Lightsail load balancer if you no longer need it. Deleting a load balancer also detaches any Lightsail instances attached to it but doesn't delete the Lightsail instances. If you enabled encrypted (HTTPS) traffic using an SSL/TLS certificate, deleting the load balancer will also permanently delete any SSL/TLS certificates associated with the load balancer.

Important

Deleting a Lightsail load balancer and its associated certificate is final and can't be undone.

1. On the Lightsail home page, choose **Networking**.
2. Choose the load balancer you want to delete.
3. Choose **Delete**.
4. Choose **Delete load balancer**.
5. Choose **Yes, delete**.

Content delivery network distributions in Amazon Lightsail

A Lightsail distribution uses a globally distributed network of servers, also known as *edge locations*, to provide faster delivery of your content to your users. To use a distribution, you first create and host your website or web application on a Lightsail instance or container service, or multiple instances attached to a Lightsail load balancer, or store your static content on a Lightsail bucket. You then create and configure a Lightsail distribution to pull, cache, and serve content from your instance, container service, load balancer, or bucket. Your instance, container service, load balancer, or bucket, also known as your distribution's *origin*, is the definitive source of your content.

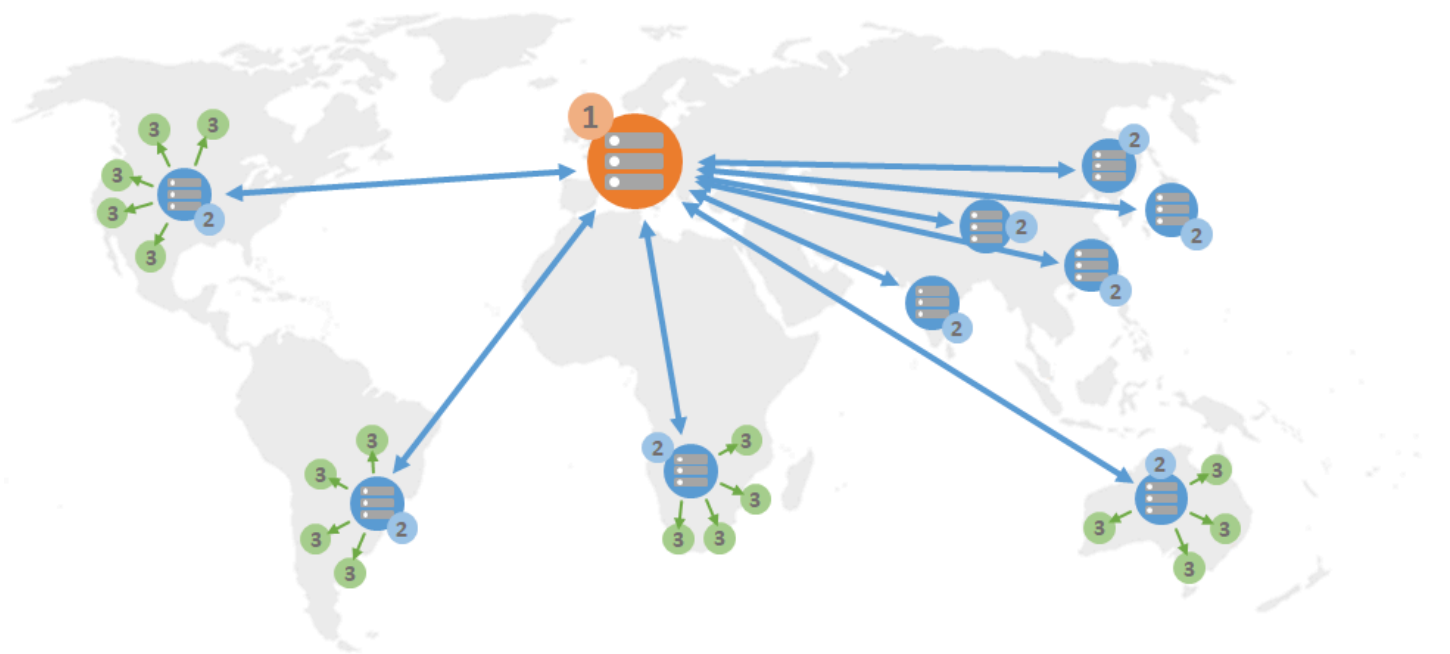
When your user requests content by visiting your website, which is being served through a distribution, the request is routed to the nearest location in terms of latency. Your distribution then performs one of the following actions:

- If the content is already being cached in the edge location, your distribution immediately serves it to your user.
- If the content is not yet being cached in that edge location, your distribution retrieves it from the specified origin, caches it, and serves it to your user.

Your content is cached in edge locations for the duration of the cache lifespan (time to live) that you specify for your distribution, so that other requests at the same location are immediately fulfilled. Your cached content is cleared from the edge location when it reaches its cache lifespan. Your distribution retrieves, caches, and serves content the next time a content request is routed to the edge location.

In the following diagram:

- 1 represents the origin of your distribution, such as a Lightsail instance or container service that is hosting your website, a load balancer with instances attached to it, or a bucket that is hosting your static content.
- 2 represents your distribution, or the edge locations that pull, cache, and serve content from your origin.
- 3 represents your users who are served content from the edge locations.

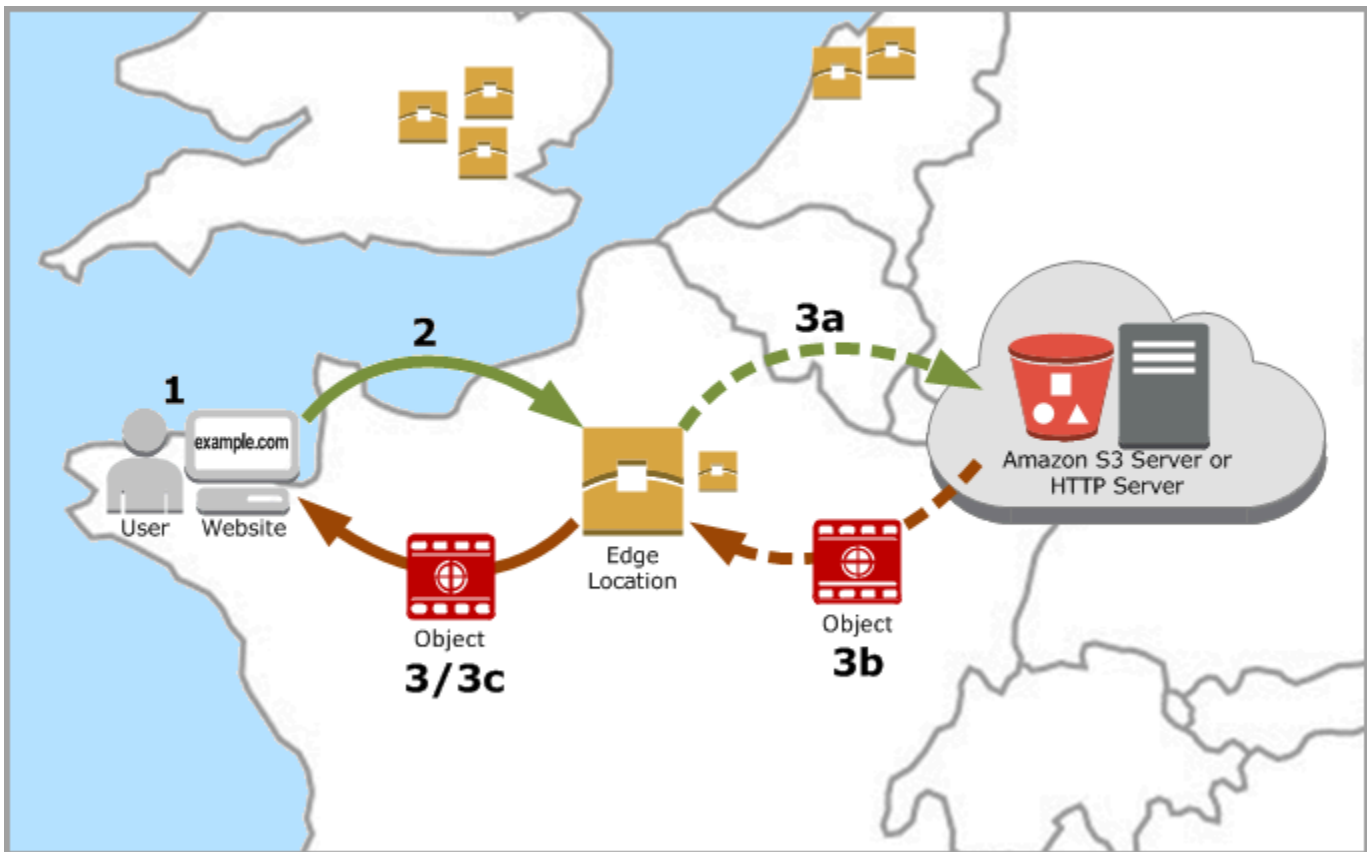
**Note**

This diagram is for illustration purpose only and doesn't show actual edge locations. For more information about edge locations, see [Edge locations and IP address ranges](#) later in this guide.

For example, if your website is hosted in France, and a person from another area of France wants to view your content, the page will load in milliseconds.

When your visitor isn't nearby, things get a little difficult.

If a person from Australia wants to view your content, the browser will have to fetch it from a server that is located in France and then show it to that user thousands of miles away. If users from different countries request the same content at the same time, the server becomes clogged with requests and takes longer to load and serve the content. This affects the speed at which the content loads for the end user.



A CDN resolves this situation by caching your website content at edge locations. This method of serving content is faster and more efficient than the traditional method of serving content from one central resource. When a viewer makes a request on your website or through your application, DNS routes the request to the location that can best serve the user's request. Your users access your content from locations that are nearby, as opposed to all of your users accessing the same central resource that may be far away.

Use cases

Deliver fast, secure websites

A Lightsail distribution speeds up the delivery of your content (for example, website pages, images, style sheets, JavaScript, and so on) to viewers *worldwide*. By using a distribution, you can take advantage of the AWS backbone network and edge servers to give your viewers a fast, safe, and reliable experience when they visit your website.

Improve your site's security

Strengthen your website and increase its performance by taking advantage of TLS termination, which reduces the load on your origin by offloading the cryptographic processing to your distribution. You can use your registered domain name together with a Lightsail SSL/TLS certificate to enable Hypertext Transfer Protocol Secure (HTTPS) for your distribution. Your users establish an encrypted HTTPS connection to your distribution, while your distribution pulls content from your origin using HTTP.

Application optimization

Easily optimize your distributions for a variety of applications, including WordPress and static websites. Using a distribution to cache and serve your content also reduces the load on your origin, because most requests are served by your distribution and not your instance, container service, load balancer, or bucket.

Configure your distribution

These are the general steps to follow to serve your website or web application using a Lightsail instance and a distribution.

1. Complete one of the following, depending on whether you want to use an instance, container service, or a bucket with your distribution.
 - **Create a Lightsail instance to host your content.** The instance serves as the origin of your distribution. The origin stores the original, definitive version of your content. For more information, see [Create an instance](#).

Attach a Lightsail static IP to your instance. Your instance's default public IP address changes if you stop and start your instance, which will break the connection between your distribution and your origin instance. A static IP does not change if you stop and start your instance. For more information, see [Create a static IP and attach it to an instance](#).

Upload your content and files to your instance. Your files, also known as *objects*, typically include web pages, images, and media files, but can be anything that can be served over HTTP.

- **Create a Lightsail container service to host your website or web application.** The container service serves as the origin of your distribution. The origin stores the original, definitive version of your content. For more information, see [Create Amazon Lightsail container services](#).

- **Create a Lightsail bucket to store your static content.** The bucket serves as the origin of your distribution. The origin stores the original, definitive version of your content. For more information, see [Create a bucket](#).

Upload files to your bucket using the Lightsail console, AWS Command Line Interface (AWS CLI), and AWS APIs. For more information about uploading files, see [Upload files to a bucket](#).

2. **(Optional) Create a Lightsail load balancer if your website being hosted on an instance requires fault tolerance.** Then attach multiple copies of your instance to your load balancer. You can configure your load balancer (with one or more instances attached to it) as the origin of your distribution, instead of configuring your instance as the origin. For more information, see [Create a load balancer and attach instances to it](#).
3. **Create a Lightsail distribution, and configure your instance, container service, load balancer, or bucket as the origin.** At the same time, you specify details such as the cache lifespan of your content, and which elements of your website or web application are cached. For more information, see [Create a distribution](#).
4. (Optional) If your distribution's origin is a WordPress instance, you must edit the WordPress configuration file in your instance to make your WordPress website work with your distribution. For more information, see [Configure your WordPress instance to work with your distribution](#).
5. **(Optional) Create a Lightsail DNS zone to manage your domain's DNS in the Lightsail console.** This allows you to easily map your domain to your Lightsail resources. For more information, see [Create a DNS zone to manage your domain's DNS records](#). Alternately, you can continue hosting your domain's DNS where it's currently being hosted.
6. **Create a Lightsail SSL/TLS certificate for your domain to use it with your distribution.** Lightsail distributions require HTTPS, so you must request an SSL/TLS certificate for your domain before you can use it with your distribution. For more information, see [Create SSL/TLS certificates for your distribution](#).
7. **Enable custom domains for your distribution to use your registered domain names with your distributions.** Enabling custom domains requires that you specify the Lightsail SSL/TLS certificate that you created for your domains. This adds your domains to your distribution and enables HTTPS. For more information, see [Enable custom domains for your distribution](#).
8. **Add an alias record to your domain's DNS to begin routing traffic for your domain to your distribution.** After you add the alias record, users who visit your domain are routed through your distribution. For more information, see [Point your domain to a distribution](#).
9. **Test that your distribution is caching your content.** For more information, see [Test your distribution](#).

Edge locations and IP address ranges

Lightsail distributions use the same edge servers and IP address ranges as Amazon CloudFront. For a list of the locations of CloudFront edge servers, see the [Amazon CloudFront Product Details page](#). For a list of the CloudFront IP ranges, see the [CloudFront global IP list](#).

Create a Lightsail content delivery network distribution

In this guide, we show you how to create an Amazon Lightsail distribution using the Lightsail console, and describe the distribution settings that you can configure. For more information about distributions, see [Content delivery network distributions](#).

Contents

- [Prerequisites](#)
- [Origin resource](#)
- [Origin protocol policy](#)
- [Caching behavior and caching presets](#)
- [Best for WordPress caching preset](#)
- [Default behavior](#)
- [Directory and file overrides](#)
- [Advanced cache settings](#)
- [Distribution plan](#)
- [Creating a distribution](#)
- [Next steps](#)

Prerequisites

Complete the following prerequisites before you get started with creating a distribution:

1. Complete one of the following, depending on whether you want to use an instance, container service, or a bucket with your distribution.
 - **Create a Lightsail instance to host your content.** The instance serves as the origin of your distribution. The origin stores the original, definitive version of your content. For more information, see [Create an instance](#).

Attach a Lightsail static IP to your instance. Your instance's default public IP address changes if you stop and start your instance, which will break the connection between your distribution and your origin instance. A static IP does not change if you stop and start your instance. For more information, see [Create a static IP and attach it to an instance](#).

Upload your content and files to your instance. Your files, also known as *objects*, typically include web pages, images, and media files, but can be anything that can be served over HTTP.

- **Create a Lightsail container service to host your website or web application.** The container service serves as the origin of your distribution. The origin stores the original, definitive version of your content. For more information, see [Creating Amazon Lightsail container services](#).
- **Create a Lightsail bucket to store your static content.** The bucket serves as the origin of your distribution. The origin stores the original, definitive version of your content. For more information, see [Create a bucket](#).

Upload files to your bucket using the Lightsail console, AWS Command Line Interface (AWS CLI), and AWS APIs. For more information about uploading files, see [Upload files to a bucket](#).

2. (Optional) Create a Lightsail load balancer if your website requires fault tolerance. Then attach multiple copies of your instance to your load balancer. You can configure your load balancer (with one or more instances attached to it) as the origin of your distribution, instead of configuring your instance as the origin. For more information, see [Create a load balancer and attach instances to it](#).

Origin resource

An *origin* is the definitive source of content for your distribution. When you create your distribution, you choose the Lightsail instance, container service, bucket, or load balancer (with one or more instances attached to it) that hosts the content of your website or web application.

Note

IPv6-only instances cannot be configured as the origin for a Lightsail content delivery network (CDN) distribution at this time.

You can choose only one origin per distribution. You can change the origin at any time after you create your distribution. For more information, see [Change the origin of your distribution](#).

Choose your origin

The origin can be an instance, with an attached static IP, that is hosting a website or application. Or it can be a load balancer that has at least one instance attached to it. Your distribution retrieves and caches content from the origin that you choose.

[Learn more about content delivery networks and origins.](#)

Select an origin from the **Oregon** (us-west-2) Region.

Instances

- Node-js-1
- LAMP_PHP_7-1
- WordPress-1

Load balancers

- LoadBalancer-1

Origin protocol policy

The origin protocol policy is the protocol policy that your distribution uses when pulling content from your origin. After you choose an origin for your distribution, you should determine if your distribution should use Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) when pulling content from your origin. If your origin is not configured for HTTPS, then you must use HTTP.

You can choose one of the following origin protocol policies for your distribution:

- **HTTP Only** - Your distribution uses only HTTP to access the origin. This is the default setting.
- **HTTPS Only** - Your distribution uses only HTTPS to access the origin.

The steps to edit your origin protocol policy are included in the [Create a distribution](#) section later in this guide.

Note

When you select a Lightsail bucket as the origin of your distribution, the **Origin protocol policy** defaults to **HTTPS only**. You cannot change the origin protocol policy when a bucket is the origin of your distribution.

Caching behavior and caching presets

A *caching preset* automatically configures the settings of your distribution for the type of content that you host on your origin. For example, choosing the **Best for static content** preset automatically configures your distribution with settings that work best with static websites. If your website is hosted on a WordPress instance, then choose the **Best for WordPress** preset to have your distribution automatically configured to work with your WordPress website.

Note

The caching preset options are not available when you select a Lightsail bucket as the origin of your distribution. We automatically apply distribution settings that are best for static content being stored in a bucket.

You can choose one of the follow caching presets for your distribution:

- **Best for static content** - This preset configures your distribution to *cache everything*. This preset is ideal if you host static content (e.g., static HTML pages) on your origin, or content that does not change for each user who visits your website. All content on your distribution is cached when you choose this preset.
- **Best for dynamic content** - This preset configures your distribution to cache nothing except the files that you specify as **Cache** in the **Directory and file overrides** section of the **Create a distribution** page. For more information, see [Directory and file overrides](#) later in this guide. This preset is ideal if you host dynamic content on your origin, or content that may change for each user who visits your website or web application.
- **Best for WordPress** - This preset configures your distribution to *cache nothing* except the files in the `wp-includes/` and `wp-content/` directories of your WordPress instance. This preset is ideal if your origin is an instance that uses the **WordPress Certified by Bitnami and Automattic**

blueprint (excluding the multisite blueprint). For more information about this preset, see [Best for WordPress caching preset](#).

Note

The **Custom settings** preset cannot be selected. It is automatically selected for you if you choose a preset but then manually modify your distribution's settings.

A caching preset can be specified only in the Lightsail console. It cannot be specified using the Lightsail API, AWS CLI, and SDKs.

Best for WordPress caching preset

When you select an instance that uses the **WordPress Certified by Bitnami and Automatic** blueprint as the origin of your distribution, Lightsail asks if you want to apply the **Best for WordPress** caching preset to your distribution. If you apply the present, then your distribution is automatically configured to work best with your WordPress website. There are no other distribution settings that you need to apply. The Best for WordPress preset to *cache nothing* except the files in the `wp-includes/` and `wp-content/` directories of your WordPress website. It also configures your distribution to clear its cache every day (cache lifespan of 1 day), allow all HTTP methods, forward only the Host header, forward no cookies, and forwards all query strings.

Important

You must edit the WordPress configuration file in your instance to make your WordPress website work with your distribution. For more information, see [Configure your WordPress instance to work with your distribution](#).

Default behavior

A *default behavior* specifies how your distribution handles content caching. The default behavior of your distribution is automatically specified for you depending on the [caching preset](#) that you select. If you select a different default behavior, then the caching preset is automatically changed to **Custom settings**.

Note

The default behavior options are not available when you select a Lightsail bucket as the origin of your distribution. We automatically apply distribution settings that are best for static content being stored in a bucket.

You can choose one of the follow default behaviors for your distribution:

- **Cache everything** - This behavior configures your distribution to cache and serve your entire website as static content. This option is ideal if your origin hosts content that doesn't change depending on who views it, or if your website does not use cookies, headers, or query strings to personalize content.
- **Cache nothing** - This behavior configures your distribution to cache only the origin files and folder paths that you specify. This option is ideal if your website or web application uses cookies, headers, and query strings to personalize content for individual users. If you select this option, you *must* specify the [directory and file path overrides](#) to cache.

Directory and file overrides

A *directory and file override* can be used to override, or add an exception to, the default behavior you selected. For example, if you chose to *cache everything*, use an override to specify a directory, file, or file type that your distribution shouldn't cache. Alternately, if you chose to *cache nothing*, use an override to specify a directory, file, or file type that your distribution should cache.

In the **Directory and file overrides** section of the page, you can specify a path to a directory or a file to cache, or not cache. Use an asterisk symbol to specify wildcard directories (path/to/assets/*), and file types (*.html, *.jpg, *.js). Directories and file paths are case-sensitive.

Note

The directory and file override options are not available when you select a Lightsail bucket as the origin of your distribution. Everything that is stored in the selected bucket is cached.

These are just a few examples of how you can specify directory and file overrides:

- Specify the following to cache all files in the document root of an Apache web server running on a Lightsail instance.

```
var/www/html/
```

- Specify the following file to cache only the index page in the document root of an Apache web server.

```
var/www/html/index.html
```

- Specify the following to cache only the .html files in the document root of an Apache web server.

```
var/www/html/*.html
```

- Specify the following to cache only the .jpg, .png, and .gif files in the images sub-directory of the document root of an Apache web server.

```
var/www/html/images/*.jpg
```

```
var/www/html/images/*.png
```

```
var/www/html/images/*.gif
```

Specify the following to cache all files in the images sub-directory of the document root of an Apache web server.

```
var/www/html/images/
```

Advanced cache settings

The *advanced settings* can be used to specify the cache lifespan of content on your distribution, the allowed HTTP methods, HTTP header forwarding, cookie forwarding, and query string forwarding. The advanced settings that you specify apply only to the directory and files that your distribution caches, including the directory and file overrides that you specify as **Cache**.

Note

The advanced cache settings are not available on the **Create distribution** page when you select a Lightsail bucket as the origin of your distribution. We automatically apply distribution settings that are best for static content being stored in a bucket. However, you can modify the advanced cache settings in the distribution management page after your distribution is created.

You can configure the following advanced settings:

Cache lifespan (TTL)

Controls the amount of time your content stays in your distribution's cache before your distribution forwards another request to your origin to determine if your content has been updated. The default value is one day. Reducing the duration allows you to better serve dynamic content. Increasing the duration means that your users get better performance because your files are more likely to be served directly from the edge location. Increasing the duration also reduces the load on your origin, because your distribution pulls content less frequently.

Note

The cache lifespan value that you specify applies only when your origin does not add HTTP headers such as `Cache-Control max-age`, `Cache-Control s-maxage`, or `Expires` to your content.

Allowed HTTP methods

Controls the HTTP methods that your distribution processes and forwards to your origin. HTTP methods indicate the desired action to be performed on the origin. For example, the GET method retrieves data from your origin, and the PUT method requests that the enclosed entity be stored on your origin.

You can choose one of the following HTTP method options for your distribution:

- **Allow GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE methods**
- **Allow the GET, HEAD, and OPTIONS methods**
- **Allow the GET and HEAD methods**

Your distribution always caches responses to the GET and HEAD requests. Your distribution also caches responses to the OPTIONS requests, if you choose to allow those requests. Your distribution does not cache responses to any other HTTP methods. For more information, see [HTTP methods](#).

Important

If you configure your distribution to allow all of the HTTP methods that are supported, you must configure your origin instance to handle all methods. For example, if you configure your distribution to allow these methods because you want to use POST, you must configure your origin server to handle DELETE requests appropriately so viewers can't delete resources that you don't want them to. For more information, search the documentation for your website or web application.

HTTP header forwarding

Controls whether your distribution caches your content based on the values of specified headers, and if so, which ones. HTTP headers carry information about the client browser, the requested page, the origin and more. For example, the Accept-Language header sends the language of the client (e.g., en-US for English), so that the origin can respond with content in the language of the client, if it's available.

You can choose one of the following HTTP header options for your distribution:

- **Forward no headers**
- **Forward only the headers I specify**

When you select **Forward no headers**, your distribution doesn't cache your content based on header values. Regardless of the option that you select, your distribution forwards certain headers to your origin and takes specific actions based on the headers that you forward. For more information about how your distribution handles header forwarding, see [HTTP request headers and distribution behavior](#).

Cookie forwarding

Controls whether your distribution forwards cookies to your origin and, if so, which ones. A cookie contains a small piece of data sent to the origin, such as information about a visitor's actions on a web page of your origin, as well as any information the visitor has provided, such as their name and interests.

You can choose one of the following cookie forwarding options for your distribution:

- **Don't forward cookies**
- **Forward all cookies**
- **Forward cookies I specify**

If you choose **Forward all cookies**, your distribution forwards all cookies regardless of how many your application uses. If you chose **Forward cookies I specify**, then enter the names of cookies that you want your distribution to forward in the text box that appears. You can specify the following wildcards when you specify cookie names:

- * matches 0 or more characters in the cookie name
- ? matches exactly one character in the cookie name

For example, suppose that a viewer's request for an object includes a cookie named `userid_`*member-number*. Where each of your users has a unique value for `member-number` (`userid_123`, `userid_124`, `userid_125`, etc.). You want your distribution to cache a separate version of the content for each member. You could accomplish this by forwarding all cookies to your origin, but the viewer requests include some cookies that you don't want your distribution to cache. You could specify the following value as a cookie name, which causes your distribution to forward all of the cookies that begin with `userid_` to your origin: `userid_*`

Query string forwarding

Controls whether your distribution forwards query strings to your origin and, if so, which ones. A query string is a part of a URL that assigns values to specified parameters. For example, the `https://example.com/over/there?name=ferret` URL contains the `name=ferret` query string. When a server receives a request for such a page, it may run a program, passing the `name=ferret` query string unchanged, to the program. The question mark is used as a separator, and is not part of the query string.

You can choose to have your distribution forward no query strings, or forward only the query strings that you specify. Choose not to forward query strings if your origin returns the same version of your content regardless of the values of query string parameters. This increases the likelihood that your distribution can serve a request from the cache, which improves performance and reduces the load on your origin. Choose to forward only the query strings that you specify if

your origin server returns different versions of your content based on one or more query string parameters.

Distribution plan

A *distribution plan* specifies the monthly data transfer quota and cost of your distribution. If your distribution transfers more data than your plan's monthly data transfer quota, you are charged an overage. For more information, see the [Lightsail pricing page](#).

To avoid an overage fee, change your distribution's current plan to a different plan that offers a greater amount of monthly data transfer before your distribution exceeds its monthly quota. You can change your distribution's plan only one time during each AWS billing cycle. For more information about changing your distributions plan after you create it, see [Change the plan of your distribution](#).

Create a distribution

Complete the following procedure to create a distribution.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose **Create distribution**.
4. In the **Choose your origin** section of the page, choose the AWS Region in which your origin resource was created.

Distributions are global resources. They can reference an origin in any AWS Region, and distribute its content globally.

5. Choose your origin. An origin can be a Lightsail instance, container service, bucket, or a load balancer (with one or more instances attached to it). For more information, see [Origin resource](#).

Important

If you choose a Lightsail container service as the origin of your distribution, Lightsail automatically adds the default domain name of your distribution as a custom domain on your container service. This enables traffic to be routed between your distribution and your container service. However, there are some circumstances in which you might need to manually add the default domain name of your distribution to your

container service. For more information, see [Add the default domain of a distribution to a container service](#).

- (Optional) To change your origin protocol policy, choose the pencil icon displayed next to the current origin protocol policy that your distribution uses. For more information, see [Origin protocol policy](#).

This option is listed in the **Choose your origin** section of the page, under the origin resource you selected for your distribution.

Note

When you select a Lightsail bucket as the origin of your distribution, the **Origin protocol policy** defaults to **HTTPS only**. You cannot change the origin protocol policy when a bucket is the origin of your distribution.



- Choose the caching behavior (also known as a caching preset) for your distribution. For more information, see [Caching behavior and caching preset](#).

Note

The caching preset options are not available when you select a Lightsail bucket as the origin of your distribution. We automatically apply distribution settings that are best for static content being stored in a bucket.

- (Optional) Choose **Show all settings** to view additional caching behavior settings for your distribution.

Note

The caching behavior settings are not available when you select a Lightsail bucket as the origin of your distribution. We automatically apply distribution settings that are best for static content being stored in a bucket.

9. (Optional) Choose the default behavior for your distribution. For more information, see [Default behavior](#).

Note

The default behavior options are not available when you select a Lightsail bucket as the origin of your distribution. We automatically apply distribution settings that are best for static content being stored in a bucket.

10. (Optional) Choose **Add path** to add a directory and file override to your distribution's caching behavior. For more information, see [Directory and file overrides](#).

Note

The directory and file override options are not available when you select a Lightsail bucket as the origin of your distribution. We automatically apply distribution settings that are best for static content being stored in a bucket.

11. (Optional) Choose the pencil icon displayed next to the advanced setting you want to edit for your distribution. For more information, see [Advanced cache settings](#).

Note

The advanced cache settings are not available on the **Create distribution** page when you select a Lightsail bucket as the origin of your distribution. We automatically apply distribution settings that are best for static content being stored in a bucket. However, you can modify the advanced cache settings in the distribution management page after your distribution is created.

12. Choose your distribution plan. For more information, see [Distribution plans](#).
13. Enter a name for your distribution.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

14. Review the cost of your distribution.

15. Choose **Create distribution**.

Your distribution is created after a few moments.

Next steps

We recommend that you complete the following next steps after your distribution is up and running.

1. If your distribution's origin is a WordPress instance, you must edit the WordPress configuration file in your instance to make your WordPress website work with your distribution. For more information, see [Configure your WordPress instance to work with your distribution](#).
2. (Optional) Create a Lightsail DNS zone to manage your domain's DNS in the Lightsail console. This allows you to easily map your domain to your Lightsail resources. For more information, see [Create a DNS zone to manage your domain's DNS records](#). Alternately, you can continue hosting your domain's DNS where it's currently being hosted.
3. Create a Lightsail SSL/TLS certificate for your domain to use it with your distribution. Lightsail distributions require HTTPS, so you must request an SSL/TLS certificate for your domain before you can use it with your distribution. For more information, see [Create SSL/TLS certificates for your distribution](#).
4. Enable custom domains for your distribution to use your domain with your distribution. Enabling custom domains requires that you specify the Lightsail SSL/TLS certificate that you created for your domain. This adds your domain to your distribution and enables HTTPS. For more information, see [Enable custom domains for your distribution](#).
5. Add an alias record to your domain's DNS to begin routing traffic for your domain to your distribution. After you add the alias record, users who visit your domain are routed through your distribution. For more information, see [Point your domain to a distribution](#).

6. Test that your distribution is caching your content. For more information, see [Test your distribution](#).

Delete a Lightsail distribution

You can delete your Amazon Lightsail distribution at any time if you're no longer using it.

Delete your distribution

Complete the following procedure to delete a distribution.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution you want to delete.
4. Choose the **Delete** tab on your distribution's management page.
5. Choose **Delete distribution** to delete your distribution.
6. Choose **Yes, delete** to confirm the deletion.

Change the caching behavior of your Lightsail distribution

A cache behavior lets you configure what is cached or isn't cached from your origin by your Amazon Lightsail distribution. For example, you can specify to cache individual directories, files, or file types from your origin. You can also specify the HTML methods and headers that are forwarded to your origin. In this guide, we show you how to change the caching behavior of your distribution. For more information about distributions, see [Content delivery network distributions](#).

Contents

- [Caching preset](#)
- [Best for WordPress caching preset](#)
- [Default behavior](#)
- [Directory and file overrides](#)
- [Advanced cache settings](#)
- [Change your distribution's cache behavior](#)

Caching preset

A *caching preset* automatically configures the settings of your distribution for the type of content that you host on your origin. For example, choosing the **Best for static content** preset automatically configures your distribution with settings that work best with static websites. If your website is hosted on a WordPress instance, then choose the **Best for WordPress** preset to have your distribution automatically configured to work with your WordPress website.

You can choose one of the follow caching presets for your distribution:

- **Best for static content** - This preset configures your distribution to *cache everything*. This preset is ideal if you host static content (e.g., static HTML pages) on your origin, or content that does not change for each user who visits your website. All content on your distribution is cached when you choose this preset.
- **Best for dynamic content** - This preset configures your distribution to cache nothing except the files that you specify as **Cache** in the **Directory and file overrides** section of the **Create a distribution** page. For more information, see [Directory and file overrides](#) later in this guide. This preset is ideal if you host dynamic content on your origin, or content that may change for each user who visits your website or web application.
- **Best for WordPress** - This preset configures your distribution to *cache nothing* except the files in the `wp-includes/` and `wp-content/` directories of your WordPress instance. This preset is ideal if your origin is an instance that uses the **WordPress Certified by Bitnami and Automattic** blueprint (excluding the multisite blueprint). For more information about this preset, see [Best for WordPress caching preset](#).

Note

The **Custom settings** preset cannot be selected. It is automatically selected for you if you choose a preset but then manually modify your distribution's settings.

A caching preset can be specified only in the Lightsail console. It cannot be specified using the Lightsail API, AWS CLI, and SDKs.

Best for WordPress caching preset

When you select an instance that uses the **WordPress Certified by Bitnami and Automattic** blueprint as the origin of your distribution, Lightsail asks if you want to apply the **Best for**

WordPress caching preset to your distribution. If you apply the present, then your distribution is automatically configured to work best with your WordPress website. There are no other distribution settings that you need to apply. The Best for WordPress preset to *cache nothing* except the files in the `wp-includes/` and `wp-content/` directories of your WordPress website. It also configures your distribution to clear its cache every day (cache lifespan of 1 day), allow all HTTP methods, forward only the Host header, forward no cookies, and forwards all query strings.

Important

You must edit the WordPress configuration file in your instance to make your WordPress website work with your distribution. For more information, see [Configure your WordPress instance to work with your distribution](#).

Default behavior

A *default behavior* specifies how your distribution handles content caching. The default behavior of your distribution is automatically specified for you depending on the [caching preset](#) that you select. If you select a different default behavior, then the caching preset is automatically changed to **Custom settings**.

You can choose one of the follow default behaviors for your distribution:

- **Cache everything** - This behavior configures your distribution to cache and serve your entire website as static content. This option is ideal if your origin hosts content that doesn't change depending on who views it, or if your website does not use cookies, headers, or query strings to personalize content.
- **Cache nothing** - This behavior configures your distribution to cache only the origin files and folder paths that you specify. This option is ideal if your website or web application uses cookies, headers, and query strings to personalize content for individual users. If you select this option, you *must* specify the [directory and file path overrides](#) to cache.

Directory and file overrides

A *directory and file override* can be used to override, or add an exception to, the default behavior you selected. For example, if you chose to *cache everything*, use an override to specify a directory, file, or file type that your distribution shouldn't cache. Alternately, if you chose to *cache nothing*, use an override to specify a directory, file, or file type that your distribution should cache.

In the **Directory and file overrides** section of the page, you can specify a path to a directory or a file to cache, or not cache. Use an asterisk symbol to specify wildcard directories (path/to/assets/*), and file types (*.html, *.jpg, *.js). Directories and file paths are case-sensitive.

These are a few examples of how you can specify directory and file overrides:

- Specify the following to cache all files in the document root of an Apache web server running on a Lightsail instance.

```
var/www/html/
```

- Specify the following to cache only the index page in the document root of an Apache web server.

```
var/www/html/index.html
```

- Specify the following to cache only the .html files in the document root of an Apache web server.

```
var/www/html/*.html
```

- Specify the following to cache only the .jpg, .png, and .gif files in the images sub-directory of the document root of an Apache web server.

```
var/www/html/images/*.jpg
```

```
var/www/html/images/*.png
```

```
var/www/html/images/*.gif
```

Specify the following to cache all files in the images sub-directory of the document root of an Apache web server.

```
var/www/html/images/
```

Advanced cache settings

The *advanced settings* can be used to specify the cache lifespan of content on your distribution, the allowed HTTP methods, HTTP header forwarding, cookie forwarding, and query string forwarding.

The advanced settings that you specify apply only to the directory and files that your distribution caches, including the directory and file overrides that you specify as **Cache**.

You can configure the following advanced settings:

Cache lifespan (TTL)

Controls the amount of time your content stays in your distribution's cache before your distribution forwards another request to your origin to determine if your content has been updated. The default value is one day. Reducing the duration allows you to better serve dynamic content. Increasing the duration means that your users get better performance because your files are more likely to be served directly from the edge location. Increasing the duration also reduces the load on your origin, because your distribution pulls content less frequently.

Note

The cache lifespan value that you specify applies only when your origin does not add HTTP headers such as `Cache-Control max-age`, `Cache-Control s-maxage`, or `Expires` to your content.

Allowed HTTP methods

Controls the HTTP methods that your distribution processes and forwards to your origin. HTTP methods indicate the desired action to be performed on the origin. For example, the GET method retrieves data from your origin, and the PUT method requests that the enclosed entity be stored on your origin.

You can choose one of the following HTTP method options for your distribution:

- **Allow GET, HEAD, OPTIONS, PUT, PATCH, POST, and DELETE methods**
- **Allow the GET, HEAD, and OPTIONS methods**
- **Allow the GET and HEAD methods**

Your distribution always caches responses to the GET and HEAD requests. Your distribution also caches responses to the OPTIONS requests, if you choose to allow those requests. Your distribution does not cache responses to any other HTTP methods.

Important

If you configure your distribution to allow all of the HTTP methods that are supported, you must configure your origin instance to handle all methods. For example, if you configure your distribution to allow these methods because you want to use POST, you must configure your origin server to handle DELETE requests appropriately so viewers can't delete resources that you don't want them to. For more information, search the documentation for your website or web application.

HTTP header forwarding

Controls whether your distribution caches your content based on the values of specified headers, and if so, which ones. HTTP headers carry information about the client browser, the requested page, the origin and more. For example, the `Accept-Language` header sends the language of the client (e.g., `en-US` for English), so that the origin can respond with content in the language of the client, if it's available.

You can choose one of the following HTTP header options for your distribution:

- **Forward no headers**
- **Forward only the headers I specify**

When you select **Forward no headers**, your distribution doesn't cache your content based on header values. Regardless of the option that you select, your distribution forwards certain headers to your origin and takes specific actions based on the headers that you forward.

Cookie forwarding

Controls whether your distribution forwards cookies to your origin and, if so, which ones. A cookie contains a small piece of data sent to the origin, such as information about a visitor's actions on a web page of your origin, as well as any information the visitor has provided, such as their name and interests.

You can choose one of the following cookie forwarding options for your distribution:

- **Don't forward cookies**
- **Forward all cookies**
- **Forward cookies I specify**

If you choose **Forward all cookies**, your distribution forwards all cookies regardless of how many your application uses. If you chose **Forward cookies I specify**, then enter the names of cookies that you want your distribution to forward in the text box that appears. You can specify the following wildcard symbols when you specify cookie names:

- * matches 0 or more characters in the cookie name
- ? matches exactly one character in the cookie name

For example, suppose that a viewer's request for an object includes a cookie named `userid_`*member-number*. Where each of your users has a unique value for `member-number` (`userid_123`, `userid_124`, `userid_125`, etc.). You want your distribution to cache a separate version of the content for each member. You could accomplish this by forwarding all cookies to your origin, but the viewer requests include some cookies that you don't want your distribution to cache. You could specify the following value as a cookie name, which causes your distribution to forward all of the cookies that begin with `userid_` to your origin: `userid_*`

Query string forwarding

Controls whether your distribution forwards query strings to your origin and, if so, which ones. A query string is a part of a URL that assigns values to specified parameters. For example, the `https://example.com/over/there?name=ferret` URL contains the `name=ferret` query string. When a server receives a request for such a page, it may run a program, passing the `name=ferret` query string unchanged, to the program. The question mark is used as a separator, and is not part of the query string.

You can choose to have your distribution forward no query strings, or forward only the query strings that you specify. Choose not to forward query strings if your origin returns the same version of your content regardless of the values of query string parameters. This increases the likelihood that your distribution can serve a request from the cache, which improves performance and reduces the load on your origin. Choose to forward only the query strings that you specify if your origin server returns different versions of your content based on one or more query string parameters.

Change your distribution's cache behavior

Complete the following procedure to change the default cache behavior of your distribution.

1. Sign in to the [Lightsail console](#).

2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which you want to change the default cache behavior.
4. Choose the **Cache** tab on your distribution's management page.
5. In the **Configure caching** section of the page, choose the caching preset for your distribution. For more information, see [Caching preset](#).
6. Choose **Change default cache behavior** to change the default behavior for your distribution. Then, choose a default behavior for your distribution. For more information, see [Default behavior](#).
7. Choose **Add path** to add a directory and file override to your distribution's caching behavior. For more information, see [Directory and file overrides](#).
8. Choose the pencil icon displayed next to the advanced setting you want to edit for your distribution. For more information, see [Advanced cache settings](#).

When you save changes to your distribution's configuration, your distribution starts to propagate the changes to all edge locations. Until your configuration is updated in an edge location, your distribution continues to serve your content from that location based on the previous configuration. After your configuration is updated in an edge location, your distribution immediately starts to serve your content from that location based on the new configuration.

Your changes don't propagate to every edge location instantaneously. When propagation is complete, the status of your distribution changes from **InProgress** to **Enabled**. While your distribution is propagating your changes, we can't determine whether a given edge location is serving your content based on the previous configuration or the new configuration.

Topics

- [Reset the cache of your Lightsail distribution](#)

Reset the cache of your Lightsail distribution

The cache lifespan (time to live) setting controls the amount of time your content stays in your Amazon Lightsail distribution's cache. You can also manually reset the cache on your distribution if you need to clear it before the cache lifespan interval. After you clear the cache, the next time a user requests content, your distribution pulls the latest version of your content from your origin and caches it. In this guide, we show you how to manually reset the cache on your distribution. For more information about distributions, see [Content delivery network distributions](#).

Reset the cache of your distribution

Complete the following procedure to reset the cache of your distribution.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which you want to reset the cache.
4. Choose the **Cache** tab on your distribution's management page.
5. Scroll to the **Reset cache** section of the page, and choose **Reset cache**.
6. At the confirmation prompt, choose **Yes, reset** to confirm that you want to reset your distribution's cache. Or choose **No, cancel** to not reset your distribution's cache.

Change the origin of your Lightsail distribution

In this guide, we show you how to change the origin of your Amazon Lightsail distribution after you create it. An origin is the definitive source of content for your distribution. When you create your distribution, you choose the Lightsail instance, Lightsail bucket, or Lightsail load balancer (with one or more instances attached to it) that hosts the content of your website or web application. For more information, see [Content delivery network distributions](#).

You can change the origin at any time after you create your distribution. When you change the origin, your distribution immediately begins replicating the change to the edge locations. Your distribution will continue to forward requests to the previous origin in a given edge location until the distribution is updated to the new origin in that edge location.

Changing the origin does not require your distribution to repopulate edge caches with content from your new origin. As long as the user requests in your website or web application have not changed, your distribution continues to serve content that is already in an edge cache until the cache lifespan for your content expires.

Origin protocol policy

The origin protocol policy is the protocol policy that your distribution uses when pulling content from your origin. After you choose an origin for your distribution, you should determine if your distribution should use Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) when pulling content from your origin. If your origin is not configured for HTTPS, then you must use HTTP.

You can choose one of the following origin protocol policies for your distribution:

- **HTTP Only** - Your distribution uses only HTTP to access the origin. This is the default setting.
- **HTTPS Only** - Your distribution uses only HTTPS to access the origin.

The steps to edit your origin protocol policy are included in the following [Change your distribution's origin](#) section of this guide.

Change your distribution's origin

Complete the following procedure to change your distribution's origin.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which you want to change the origin.
4. Choose the **Details** tab on your distribution's management page, and scroll to the **Choose your origin** section of the page.

The **Select your origin** section of the page displays your distribution's current origin.

5. Choose **Change origin**.
6. Choose the AWS Region in which your origin resource was created.

Distributions are global resources. They can reference an origin in any AWS Region, and distribute its content globally.

7. Choose your origin. An origin can be an instance, bucket, or a load balancer (with one or more instances attached to it).
8. Choose **Save** to update your distribution with your new origin.

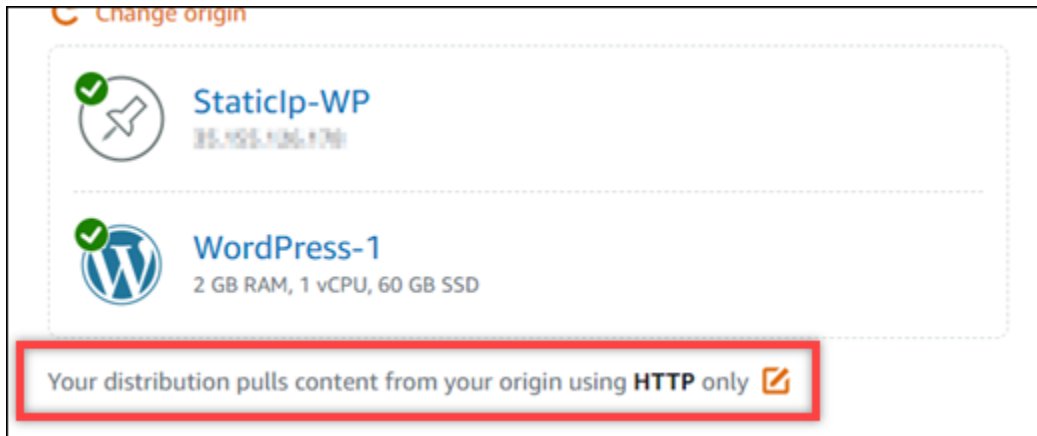
After you choose an origin for your distribution, you should determine if your distribution should use Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) when pulling content from your origin.

9. (Optional) To change your origin protocol policy, choose the pencil icon displayed next to the current origin protocol policy that your distribution uses. For more information, see [Origin protocol policy](#).

This option is listed in the **Choose your origin** section of the page, under the origin resource you selected for your distribution.

Note

When you select a Lightsail bucket as the origin of your distribution, the **Origin protocol policy** defaults to **HTTPS only**. You cannot change the origin protocol policy when a bucket is the origin of your distribution.



10. Choose **HTTP only** or **HTTPS only**, then choose **Save** to save the origin protocol policy.

When you save changes to your distribution's configuration, your distribution starts to propagate the changes to all edge locations. Until your configuration is updated in an edge location, your distribution continues to serve your content from that location based on the previous configuration. After your configuration is updated in an edge location, your distribution immediately starts to serve your content from that location based on the new configuration.

Your changes don't propagate to every edge location instantaneously. When propagation is complete, the status of your distribution changes from **InProgress** to **Enabled**. While your distribution is propagating your changes, we can't determine whether a given edge location is serving your content based on the previous configuration or the new configuration.

Change the plan of your Lightsail distribution

When you create a Amazon Lightsail distribution, you choose a distribution plan that specifies the monthly data transfer quota and cost of your distribution. If your distribution transfers more data than your plan's monthly data transfer quota, you are charged an overage. For more information about overage pricing, see the [Lightsail pricing page](#).

To avoid an overage fee, change your distribution's current plan to a different plan that offers a greater amount of monthly data transfer before your distribution exceeds its monthly quota. You can change your distribution's plan only one time during each AWS billing cycle. In this guide, we show you how to change your distribution's plan.

For more information about distributions, see [Content delivery network distributions](#).

Change your distribution plan

Complete the following procedure to change your distribution's plan.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which you want to view the current monthly data transfer.
4. Choose the **Details** tab on your distribution's management page.
5. In the **Data transfer** section of the page, choose **Change distribution plan**.
6. At the confirmation prompt, choose **Yes, change** to confirm that you want to change your distribution's plan.
7. On the next prompt, choose the new plan for your distribution, and choose **Select plan**.
8. On the next prompt, choose **Yes, apply** to confirm that you want to apply the new plan to your distribution. Or choose **No, go back** to not apply the new plan to your distribution.

Custom domains for your Lightsail distribution

Enable custom domains for your Amazon Lightsail distribution to use your registered domain names with your distribution. Before you enable custom domains, your distribution accepts traffic only for the default domain that is associated with your distribution when you first create it (e.g., `123456abcdef.cloudfront.net`). When you enable custom domains, you must choose the Lightsail SSL/TLS certificate that you created for the domains that you want to use with your distribution. After you enable custom domains, your distribution accepts traffic for all of the domains that are associated with the certificate that you chose.

Important

Only one certificate can be in use at a time per distribution. If you disable custom domains on your distribution, your distribution is no longer able to handle HTTPS traffic for your registered domain until you enable custom domains again.

The domain names associated with the SSL/TLS certificate cannot be in use by another distribution across all Amazon Web Services (AWS) accounts, including distributions on the Amazon CloudFront service. You will be able to create the certificate for the domains, but you will not be able to use it with your distribution.

For more information about distributions, see [Content delivery network distributions](#).

Prerequisites

Before you get started, you need to create a Lightsail distribution. For more information, see [Create a distribution](#).

You also should have created and validated an SSL/TLS certificate for your distribution. For more information, see [Create SSL/TLS certificates for your distribution](#) and [Validate SSL/TLS certificates for your distribution](#).

Enable custom domains for your distribution

Complete the following procedure to enable custom domains for your distribution.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which want to enable custom domains.
4. Choose the **Custom domains** tab on your distribution's management page.
5. Choose **Attach certificate**.

If you have no certificates, then you must first create and validate an SSL/TLS certificate for your domains, before you can attach it to your distribution. For more information, see [Create SSL/TLS certificates for your distribution](#).

6. In the dropdown menu that appears, select a valid certificate for the domain(s) that you want to use with your distribution.

7. Verify the certificate information is correct, then choose **Attach**.
8. The distribution's **Status** will change to **Updating**. After the status changes to **Enabled**, the certificate's domain will appear in the **Custom domains** section.
9. Choose **Add domain assignment** to point the domain to your distribution.
10. Verify the certificate and DNS information are correct, then choose **Add assignment**. After a few moments, traffic for the domain that you selected will begin to be accepted by your distribution.

Topics

- [Point a domain to your Lightsail distribution](#)
- [Change the custom domain for your Lightsail distribution](#)
- [Disable custom domains for your Lightsail distribution](#)
- [Add the default domain of a distribution to a Lightsail container service](#)

Point a domain to your Lightsail distribution

You must point your registered domain names to your Amazon Lightsail distribution after you enabled custom domains for your distribution. You do this by adding an alias record to the DNS zone of each of the domains specified on the certificate that you're using with your distribution. All of the records that you add should point to the default domain (e.g., `123456abcdef.cloudfront.net`) of your distribution.

In this guide, we provide you with the procedure to point your domains to your distribution using a Lightsail DNS zone. The procedure to point your domains to your distribution using a different DNS hosting provider, like Domain.com or GoDaddy, may be similar. For more information about Lightsail DNS zones, see [DNS](#).

For more information about distributions, see [Create a distribution](#).

Contents

- [Step 1: Complete the prerequisite](#)
- [Step 2: Get the default domain of your distribution](#)
- [Step 3: Add a record to your domain's DNS zone](#)

Step 1: Complete the prerequisite

Before you get started, you should enable custom domains for your Lightsail distribution. For more information, see [Enable custom domains for your distribution](#).

Step 2: Get the default domain of your distribution

Complete the following procedure to get default domain name of your distribution, which you specify when you add an alias record to the DNS of your domain.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which want get the default domain name.
4. In the header section of your distribution's management page, make note of your distribution's default domain name. Your distribution's default domain name is similar to `123456abcdef.cloudfront.net`.

You must add this value as part of an alias record in the DNS of your domains. We recommend that you copy and paste this value into a text file that you can refer to later. Continue to the next [Step 3: Add a record to your domain's DNS zone](#) section of this tutorial.

Step 3: Add a record to your domain's DNS zone

Complete the following procedure to add a record to your domain's DNS zone.

1. On the Lightsail home page, choose the **Domains & DNS** tab.
2. Under the **DNS zones** section of the page, choose the domain name to which you want to add the record that will direct traffic for your domain to your distribution.
3. Choose the **DNS records** tab. Then, choose **Add record**.
4. Complete one of the following steps depending on the type of domain that you want to point to your distribution:
 - Choose an address (A) record to point an apex domain (e.g., `example.com`) to your distribution.

If an A record for the apex of your domain is already present in your DNS zone, then you will need to edit that existing record instead of adding another A record.

- Choose a canonical name (CNAME) to point a sub domain, such as `website.example.com`, to your distribution.
5. If you're adding an A record, then in the **Resolves to** text box choose the name of your distribution. If you're adding a CNAME record, then in the **Maps to** text box enter the default domain name of your distribution.

Note

When you add an A record to your DNS zone, and choose the name of your distribution, you are in fact adding an alias record, which is different than an address record. Lightsail makes it easy for you to add alias records without the additional steps that are typically required at other DNS hosting providers.

6. Choose the save icon to save the record to your DNS zone.

Repeat these steps to add additional DNS records for domains on your certificate that you are using with your distribution. Allow time for changes to propagate through the Internet's DNS. After a few minutes, you should see if your domain is pointing to your distribution. You should also test your distribution. For more information, see the following [Test your distribution](#).

Change the custom domain for your Lightsail distribution

You can change the custom domains used by your Amazon Lightsail distribution to another domain or set of domains. To do so, you must first create a new SSL/TLS certificate for the domains that you want to use with your distribution. For more information, see [Create SSL/TLS certificates for your distribution](#). After the new certificate is validated, you swap the old certificate for the new one, thereby changing the custom domains for your distribution.

For more information about distributions, see [Create a distribution](#).

Change custom domains for your distribution

Complete the following procedure to change the custom domains for your distribution.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which you want to change the custom domains.

4. Choose the **Custom domains** tab on your distribution's management page.
5. Detach the SSL/TLS certificate that is currently attached to the distribution.

The status of the distribution will change to **In progress**.

6. After the distribution's status changes back to **Enabled**, choose **Attach certificate**.
7. In the dropdown menu that appears, select a valid certificate for the domain(s) that you want to use with your distribution.
8. Verify the certificate information is correct, then choose **Attach**.
9. Add a domain assignment to the DNS of your domain to point the domain to your distribution.

The distribution's **Status** will change to **Updating**. After the status changes to **Ready**, the certificate's domain will appear in the **Custom domains** section. Choose **Add domain assignment** to point the domain to your distribution.

10. Choose **Add assignment**. After a few moments, traffic for the domain that you selected will begin to be accepted by your distribution.
11. Choose **Save**.

Disable custom domains for your Lightsail distribution

Disable custom domains for your Amazon Lightsail distribution to stop using your registered domain names with your distribution. After you disable custom domains, your distribution accepts traffic only for the default domain that is associated with your distribution when you first create it (e.g., `123456abcdef.cloudfront.net`), and traffic for the previously associated custom domains will see a 403 error.

For more information about distributions, see [Content delivery network distributions](#).

Disable custom domains for your distribution

Complete the following procedure to disable custom domains for your distribution.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which want to disable custom domains.
4. Choose the **Custom domains** tab on your distribution's management page.

The **Custom domains** page displays the SSL/TLS certificates currently attached to your distribution, if any.

5. Choose one of the following options:
 1. Choose **Configure distribution domains** to either deselect domains that were previously selected, or to select more domains that are associated to the distribution.
 2. Choose **Detach** to detach the certificate from the distribution, and remove all of its associated domains.
6. Your request to disable custom domains is submitted, and the status of your distribution is changed to **In progress**. After a while, the status of your distribution changes to **Enabled**.

After you disable custom domains, your distribution accepts traffic only for the default domain that is associated with your distribution when you first create it (e.g., `123456abcdef.cloudfront.net`), and traffic for the previously associated custom domains will see a 403 error. You should update the DNS records of the domains so that traffic for those domains is directed to another resource.

Add the default domain of a distribution to a Lightsail container service

You can choose an Amazon Lightsail container service as the origin of a content delivery network (CDN) distribution. The distribution then caches and serves the website or web application hosted on your container service. If you're using a Lightsail distribution with your Lightsail container service, Lightsail automatically adds the default domain name of your distribution as a custom domain on your container service. This enables traffic to be routed between your distribution and your container service. However, you *must* perform the steps outlined in this guide to manually add the default domain name of your distribution to your container service under the following circumstances:

- If something goes wrong and your distribution's default domain name is not automatically added to your container service.
- If you're using a distribution other than a Lightsail distribution with your container service.

You can manually add the default domain name of your distribution to your container service only by using the AWS Command Line Interface (AWS CLI). For more information about container services, see [Container services](#). For more information about distributions, see [Object storage](#).

Add the default domain of a distribution to a container service

Complete the following procedure to add the default domain of a distribution to a container service in Lightsail using the AWS Command Line Interface (AWS CLI). You do this by using the `update-container-service` command. For more information, see [update-container-service](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter one of the following commands to add the default domain of a distribution to a container service.

Note

If you added a custom domain to your container service, then you will need to specify both your custom domain and the default domain of your distribution.

No custom domain is configured on the container service:

```
aws lightsail update-container-service --service-name ContainerServiceName --public-domain-names '{"_": ["DistributionDefaultDomain"]}'
```

One or more custom domains are configured on the container service:

```
aws lightsail update-container-service --service-name ContainerServiceName --public-domain-names '{"CertificateName": ["ExistingCustomDomain"], "_": ["DistributionDefaultDomain"]}'
```

In the command, replace the following example text with your own:

- *ContainerServiceName* - The name of the Lightsail container service that was specified as the origin of the distribution.

- *DistributionDefaultDomain* - The default domain of the distribution that is using the container service as the origin. For example, `example123.cloudfront.net`.
- *CertificateName* - The name of the Lightsail certificate of the custom domains that are currently attached to the container service, if any. If there are no custom domains attached to the container service, then use the command labeled as **No custom domain is configured on the container service**.
- *DistributionDefaultDomain* - The custom domain currently attached to the container service.

Examples:

- **No custom domain is configured on the container service:**

```
aws lightsail update-container-service --service-name ContainerServiceName --public-domain-names '{"_": ["example123.cloudfront.net"]}'
```

- **One or more custom domains are configured on the container service:**

```
aws lightsail update-container-service --service-name ContainerServiceName --public-domain-names '{"example-com": ["example.com"], "_": ["example123.cloudfront.net"]}'
```

Lightsail distribution request and response behaviors

In this guide, we describe the way your Amazon Lightsail distribution behaves when processing and forwarding requests to your origin, and processing responses from your origin. For more information about distributions, see [Content delivery network distributions](#).

Topics

- [How your distribution processes and forwards requests to your origin](#)
- [How your distribution processes responses from your origin](#)

How your distribution processes and forwards requests to your origin

This section contains information about how your distribution processes viewer requests and forwards the requests to your origin.

Contents

- [Authentication](#)
- [Caching duration](#)
- [Client IP addresses](#)
- [Client-side SSL authentication](#)
- [Compression](#)
- [Conditional requests](#)
- [Cookies](#)
- [Cross-origin resource sharing \(CORS\)](#)
- [Encryption](#)
- [GET requests that include a body](#)
- [HTTP methods](#)
- [HTTP request headers and distribution behavior](#)
- [HTTP version](#)
- [Maximum length of a request and maximum length of a URL](#)
- [OCSP stapling](#)
- [Persistent connections](#)
- [Protocols](#)
- [Query strings](#)
- [Origin connection timeout and attempts](#)
- [Origin response timeout](#)
- [Simultaneous requests for the same object \(traffic spikes\)](#)
- [User-agent header](#)

Authentication

For DELETE, GET, HEAD, PATCH, POST, and PUT requests, if you configure your distribution to forward the `Authorization` header to your origin, you can configure your origin server to request client authentication.

For OPTIONS requests, you can configure your origin server to request client authentication only if you use the following distribution settings:

- Configure your distribution to forward the Authorization header to your origin.
- Configure your distribution to not cache the response to OPTIONS requests.

You can configure your distribution to forward requests to your origin using either HTTP or HTTPS.

Caching duration

To control how long your objects stay in your distribution's cache before your distribution forwards another request to your origin, you can:

- Configure your origin to add a Cache-Control or an Expires header field to each object.
- Use the default value of 1 day for the cache lifespan (TTL).

For more information, see [distribution advanced settings](#).

Client IP addresses

If a viewer sends a request to your distribution and does not include an X-Forwarded-For request header, your distribution gets the IP address of the viewer from the TCP connection, adds an X-Forwarded-For header that includes the IP address, and forwards the request to the origin. For example, if your distribution gets the IP address 192.0.2.2 from the TCP connection, it forwards the following header to the origin:

```
X-Forwarded-For: 192.0.2.2
```

If a viewer sends a request to your distribution and includes an X-Forwarded-For request header, your distribution gets the IP address of the viewer from the TCP connection, appends it to the end of the X-Forwarded-For header, and forwards the request to the origin. For example, if the viewer request includes X-Forwarded-For: 192.0.2.4,192.0.2.3 and your distribution gets the IP address 192.0.2.2 from the TCP connection, it forwards the following header to the origin:

```
X-Forwarded-For: 192.0.2.4,192.0.2.3,192.0.2.2
```

Some applications, such as load balancers, web application firewalls, reverse proxies, intrusion prevention systems, and API Gateway, append the IP address of the distribution edge server that forwarded the request onto the end of the X-Forwarded-For header. For example, if your

distribution includes `X-Forwarded-For: 192.0.2.2` in a request that it forwards to ELB and if the IP address of the distribution edge server is `192.0.2.199`, the request that your instance receives contains the following header:

```
X-Forwarded-For: 192.0.2.2,192.0.2.199
```

Note

The `X-Forwarded-For` header contains IPv4 addresses (such as `192.0.2.44`) and IPv6 addresses (such as `2001:0db8:85a3:0000:0000:8a2e:0370:7334`).

Client-side SSL authentication

Lightsail distributions don't support client authentication with client-side SSL certificates. If an origin requests a client-side certificate, your distribution drops the request.

Compression

Lightsail distributions forward requests that have the `Accept-Encoding` field values `"identity"` and `"gzip"`.

Conditional requests

When your distribution receives a request for an object that has expired from an edge cache, it forwards the request to your origin either to get the latest version of the object or to get confirmation from the origin that the distribution edge cache already has the latest version. Typically, when the origin last sent the object to your distribution, it included an `ETag` value, a `LastModified` value, or both values in the response. In the new request that your distribution forwards to your origin, your distribution adds one or both of the following:

- An `If-Match` or `If-None-Match` header that contains the `ETag` value for the expired version of the object.
- An `If-Modified-Since` header that contains the `LastModified` value for the expired version of the object.

The origin uses this information to determine whether the object has been updated and, therefore, whether to return the entire object to your distribution or to return only an HTTP 304 status code (not modified).

Cookies

You can configure your distribution to forward cookies to your origin. For more information, see [distribution advanced settings](#).

Cross-origin resource sharing (CORS)

If you want your distribution to respect cross-origin resource sharing settings, configure your origin to forward the `Origin` header to your origin.

Encryption

You can require viewers to connect to your distribution using HTTPS and require your distribution to forward requests to your origin by using HTTP or HTTPS.

Your distribution forwards HTTPS requests to your origin using the SSLv3, TLSv1.0, TLSv1.1, and TLSv1.2 protocols. Other versions of SSL and TLS are not supported.

GET requests that include a body

If a viewer GET request includes a body, your distribution returns an HTTP status code 403 (Forbidden) to the viewer.

HTTP methods

If you configure your distribution to allow all of the HTTP methods that it supports, your distribution accepts the following requests from viewers and forwards them to your origin:

- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT

Your distribution always caches responses to GET and HEAD requests. You can also configure your distribution to cache responses to OPTIONS requests. Your distribution does not cache responses to requests that use the other methods.

For information about configuring whether your origin processes these methods, see the documentation for your origin.

Important

If you configure your distribution to accept and forward to your origin all of the HTTP methods that it supports, configure your origin server to handle all methods. For example, if you configure your distribution to accept and forward these methods because you want to use POST, you must configure your origin server to handle DELETE requests appropriately so viewers can't delete resources that you don't want them to. For more information, see the documentation for your HTTP server.

HTTP request headers and distribution behavior

The following list contains the HTTP request headers that you can forward to your origin (with the exceptions that are noted). For each header, the list includes information about the following:

- **Supported** - Whether you can configure your distribution to cache objects based on header values for that header.

You can configure your distribution to cache objects based on values in the Date and User-Agent headers, but we don't recommend it. These headers have many possible values, and caching based on their values would cause your distribution to forward significantly more requests to your origin.

- **Behavior if you not configured** - The behavior of your distribution if you don't configure it to forward the header to your origin, which causes your distribution to cache your objects based on header values.

- **Header** - Other-defined headers

Supported - Yes

Behavior if not configured - Your distribution forwards the headers to your origin.

- **Header** - Accept

Supported - Yes

Behavior if not configured - Your distribution removes the header.

- **Header - Accept-Charset**

Supported - Yes

Behavior if not configured - Your distribution removes the header.

- **Header - Accept-Encoding**

Supported - Yes

Behavior if not configured - If the value contains `gzip`, Your distribution forwards `Accept-Encoding: gzip` to your origin. If the value does not contain `gzip`, Your distribution removes the `Accept-Encoding` header field before forwarding the request to your origin.

- **Header - Accept-Language**

Supported - Yes

Behavior if not configured - Your distribution removes the header.

- **Header - Authorization**

Supported - Yes

Behavior if not configured:

- GET and HEAD requests – Your distribution removes the `Authorization` header field before forwarding the request to your origin.
- OPTIONS requests – Your distribution removes the `Authorization` header field before forwarding the request to your origin if you configure your distribution to cache responses to OPTIONS requests.

Your distribution forwards the `Authorization` header field to your origin if you do not configure your distribution to cache responses to OPTIONS requests.

- DELETE, PATCH, POST, and PUT requests – Your distribution does not remove the header field before forwarding the request to your origin.

- **Header - Cache-Control**

Supported - No

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header - CloudFront-Forwarded-Proto**

How your distribution processes and forwards requests to your origin

Supported - Yes

Behavior if not configured - Your distribution does not add the header before forwarding the request to your origin.

- **Header** - CloudFront-Is-Desktop-Viewer

Supported - Yes

Behavior if not configured - Your distribution does not add the header before forwarding the request to your origin.

- **Header** - CloudFront-Is-Mobile-Viewer

Supported - Yes

Behavior if not configured - Your distribution does not add the header before forwarding the request to your origin.

- **Header** - CloudFront-Is-Tablet-Viewer

Supported - Yes

Behavior if not configured - Your distribution does not add the header before forwarding the request to your origin.

- **Header** - CloudFront-Viewer-Country

Supported - Yes

Behavior if not configured - Your distribution does not add the header before forwarding the request to your origin.

- **Header** - Connection

Supported - No

Behavior if not configured - Your distribution replaces this header with `Connection: Keep-Alive` before forwarding the request to your origin.

- **Header** - Content-Length

Supported - No

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header - Content-MD5**

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header - Content-Type**

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header - Cookie**

Supported - No

Behavior if not configured - If you configure Your distribution to forward cookies, it will forward the Cookie header field to your origin. If you don't, your distribution removes the Cookie header field.

- **Header - Date**

Supported - Yes, but not recommended

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header - Expect**

Supported - Yes

Behavior if not configured - Your distribution removes the header.

- **Header - From**

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header - Host**

Supported - Yes

Behavior if not configured - Your distribution sets the value to the domain name of the origin that is associated with the requested object.

- **Header - If-Match**

How your distribution processes and forwards requests to your origin

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - If-Modified-Since

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - If-None-Match

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - If-Range

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - If-Unmodified-Since

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - Max-Forwards

Supported - No

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - Origin

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - Pragma

Supported - No

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - Proxy-Authenticate

Supported - No

Behavior if not configured - Your distribution removes the header.

- **Header** - Proxy-Authorization

Supported - No

Behavior if not configured - Your distribution removes the header.

- **Header** - Proxy-Connection

Supported - No

Behavior if not configured - Your distribution removes the header.

- **Header** - Range

Supported - Yes, by default

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - Referer

Supported - Yes

Behavior if not configured - Your distribution removes the header.

- **Header** - Request-Range

Supported - No

Behavior if not configured - >Your distribution forwards the header to your origin.

- **Header** - TE

Supported - No

Behavior if not configured - Your distribution removes the header.

- **Header** - Trailer

Supported - No

Behavior if not configured - Your distribution removes the header.

- **Header** - Transfer-Encoding

Supported - No

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - Upgrade

Supported - No (except for WebSocket connections)

Behavior if not configured - Your distribution removes the header, unless you've established a WebSocket connection.

- **Header** - User-Agent

Supported - Yes, but not recommended

Behavior if not configured - Your distribution replaces the value of this header field with Amazon CloudFront.

- **Header** - Via

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - Warning

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - X-Amz-Cf-Id

Supported - No

Behavior if not configured - Your distribution adds the header to the viewer request before forwarding the request to your origin. The header value contains an encrypted string that uniquely identifies the request.

- **Header** - X-Edge-*

Supported - No

Behavior if not configured - Your distribution removes all X-Edge-* headers.

- **Header** - X-Forwarded-For

Supported - Yes

Behavior if not configured - Your distribution forwards the header to your origin.

- **Header** - X-Forwarded-Proto

Supported - No

Behavior if not configured - Your distribution removes the header.

- **Header** - X-Real-IP

Supported - No

Behavior if not configured - Your distribution removes the header.

HTTP version

Your distribution forwards requests to your origin using HTTP/1.1.

Maximum length of a request and maximum length of a URL

The maximum length of a request, including the path, the query string (if any), and headers, is 20,480 bytes.

Your distribution constructs a URL from the request. The maximum length of this URL is 8192 bytes.

If a request or a URL exceeds these maximums, your distribution returns HTTP status code 413, Request Entity Too Large, to the viewer, and then terminates the TCP connection to the viewer.

OCSP stapling

When a viewer submits an HTTPS request for an object, either your distribution or the viewer must confirm with the certificate authority (CA) that the SSL certificate for the domain has not been revoked. OCSP stapling speeds up certificate validation by allowing your distribution to validate the certificate and to cache the response from the CA, so the client doesn't need to validate the certificate directly with the CA.

The performance improvement of OCSP stapling is more pronounced when your distribution receives numerous HTTPS requests for objects in the same domain. Each server in a distribution edge location must submit a separate validation request. When your distribution receives a lot of

HTTPS requests for the same domain, every server in the edge location soon has a response from the CA that it can "staple" to a packet in the SSL handshake; when the viewer is satisfied that the certificate is valid, your distribution can serve the requested object. If your distribution doesn't get much traffic in an edge location, new requests are more likely to be directed to a server that hasn't validated the certificate with the CA yet. In that case, the viewer separately performs the validation step and the distribution server serves the object. That distribution server also submits a validation request to the CA, so the next time it receives a request that includes the same domain name, it has a validation response from the CA.

Persistent connections

When your distribution gets a response from your origin, it tries to maintain the connection for several seconds in case another request arrives during that period. Maintaining a persistent connection saves the time required to re-establish the TCP connection and perform another TLS handshake for subsequent requests.

Protocols

Your distribution forwards HTTP or HTTPS requests to the origin server based on value of the **Origin protocol policy** field in the Lightsail console . In the Lightsail console, the options are **HTTP only**, and **HTTPS only**.

If you specify **HTTP Only** or **HTTPS Only**, your distribution forwards requests to your origin using the specified protocol, regardless of the protocol in the viewer request.

Important

If your distribution forwards a request to your origin using the HTTPS protocol, and if the origin server returns an invalid certificate or a self-signed certificate, your distribution drops the TCP connection.

Query strings

You can configure whether your distribution forwards query string parameters to your origin.

Origin connection timeout and attempts

By default, your distribution waits as long as 30 seconds (3 attempts of 10 seconds each) before returning an error response to the viewer.

Origin response timeout

The *origin response timeout*, also known as the *origin read timeout* or *origin request timeout*, applies to both of the following:

- The amount of time, in seconds, that your distribution waits for a response after forwarding a request to the origin.
- The amount of time, in seconds, that your distribution waits after receiving a packet of a response from the origin and before receiving the next packet.

Your distribution's behavior depends on the HTTP method of the viewer request:

- GET and HEAD requests – If the origin doesn't respond or stops responding within the duration of the response timeout, your distribution drops the connection. If the specified number of origin connection attempts is more than 1, your distribution tries again to get a complete response. Your distribution tries up to 3 times, as determined by the value of the *origin connection attempts* setting. If the origin doesn't respond during the final attempt, your distribution doesn't try again until it receives another request for content on the same origin.
- DELETE, OPTIONS, PATCH, PUT, and POST requests – If the origin doesn't respond within 30 seconds, your distribution drops the connection and doesn't try again to contact the origin. The client can resubmit the request if necessary.

Simultaneous requests for the same object (traffic spikes)

When a distribution edge location receives a request for an object and either the object isn't currently in the cache or the object has expired, your distribution immediately sends the request to your origin. If there's a traffic spike—if additional requests for the same object arrive at the edge location before your origin responds to the first request—your distribution pauses briefly before forwarding additional requests for the object to your origin. Typically, the response to the first request will arrive at the distribution edge location before the response to subsequent requests. This brief pause helps to reduce unnecessary load on your origin server. If additional requests are not identical because, for example, you configured your distribution to cache based on request headers or cookies, your distribution forwards all of the unique requests to your origin.

User-agent header

If you want your distribution to cache different versions of your objects based on the device that a user is using to view your content, we recommend that you configure your distribution to forward one or more of the following headers to your origin:

- `CloudFront-Is-Desktop-Viewer`
- `CloudFront-Is-Mobile-Viewer`
- `CloudFront-Is-SmartTV-Viewer`
- `CloudFront-Is-Tablet-Viewer`

Based on the value of the `User-Agent` header, your distribution sets the value of these headers to `true` or `false` before forwarding the request to your origin. If a device falls into more than one category, more than one value might be `true`. For example, for some tablet devices, your distribution might set both `CloudFront-Is-Mobile-Viewer` and `CloudFront-Is-Tablet-Viewer` to `true`.

You can configure your distribution to cache objects based on values in the `User-Agent` header, but we don't recommend it. The `User-Agent` header has many possible values, and caching based on those values would cause your distribution to forward significantly more requests to your origin.

If you do not configure your distribution to cache objects based on values in the `User-Agent` header, your distribution adds a `User-Agent` header with the following value before it forwards a request to your origin:

```
User-Agent = Amazon CloudFront
```

Your distribution adds this header regardless of whether the request from the viewer includes a `User-Agent` header. If the request from the viewer includes a `User-Agent` header, your distribution removes it.

How your distribution processes responses from your origin

This section contains information about how your distribution processes responses from your origin.

Contents

- [100-Continue responses](#)
- [Caching](#)
- [Canceled requests](#)
- [Content negotiation](#)
- [Cookies](#)
- [Dropped TCP connections](#)
- [HTTP response headers that your distribution removes or replaces](#)
- [Maximum file size](#)
- [Origin unavailable](#)
- [Redirects](#)
- [Transfer encoding](#)

100-Continue responses

Your origin cannot send more than one 100-Continue response to your distribution. After the first 100-Continue response, your distribution expects an HTTP 200 OK response. If your origin sends another 100-Continue response after the first one, your distribution returns an error.

Caching

- Ensure that your origin sets valid and accurate values for the Date and Last-Modified header fields.
- If requests from viewers include the If-Match or If-None-Match request header fields, set the ETag response header field. If you do not specify an ETag value, your distribution ignores subsequent If-Match or If-None-Match headers.
- Your distribution normally respects a Cache-Control: no-cache header in the response from the origin. For an exception, see [Simultaneous requests for the same object \(traffic spikes\)](#).

Canceled requests

If an object is not in the edge cache, and if a viewer terminates a session (for example, closes a browser) after your distribution gets the object from your origin but before it can deliver the requested object, your distribution does not cache the object in the edge location.

Content negotiation

If your origin returns `Vary: *` in the response, and if the value of **Minimum TTL** for the corresponding cache behavior is **0**, your distribution caches the object but still forwards every subsequent request for the object to the origin to confirm that the cache contains the latest version of the object. Your distribution doesn't include any conditional headers, such as `If-None-Match` or `If-Modified-Since`. As a result, your origin returns the object to your distribution in response to every request.

If your origin returns `Vary: *` in the response, and if the value of **Minimum TTL** for the corresponding cache behavior is any other value, CloudFront processes the `Vary` header as described in [HTTP response headers that your distribution removes or replaces](#).

Cookies

If you enable cookies for a cache behavior, and if the origin returns cookies with an object, your distribution caches both the object and the cookies. Note that this reduces cache-ability for an object.

Dropped TCP connections

If the TCP connection between your distribution and your origin drops while your origin is returning an object to your distribution, your distribution's behavior depends on whether your origin included a `Content-Length` header in the response:

- **Content-Length header** – Your distribution returns the object to the viewer as it gets the object from your origin. However, if the value of the `Content-Length` header doesn't match the size of the object, your distribution doesn't cache the object.
- **Transfer-Encoding: Chunked** – Your distribution returns the object to the viewer as it gets the object from your origin. However, if the chunked response is not complete, your distribution does not cache the object.
- **No Content-Length header** – Your distribution returns the object to the viewer and caches it, but the object may not be complete. Without a `Content-Length` header, your distribution cannot determine whether the TCP connection was dropped accidentally or on purpose.

We recommend that you configure your HTTP server to add a `Content-Length` header to prevent your distribution from caching partial objects.

HTTP response headers that your distribution removes or replaces

Your distribution removes or updates the following header fields before forwarding the response from your origin to the viewer:

- **Set-Cookie** – If you configure your distribution to forward cookies, it will forward the Set-Cookie header field to clients.
- **Trailer**
- **Transfer-Encoding** – If your origin returns this header field, your distribution sets the value to chunked before returning the response to the viewer.
- **Upgrade**
- **Vary** – Note the following:
 - If you configure your distribution to forward any of the device-specific headers to your origin (`CloudFront-Is-Desktop-Viewer`, `CloudFront-Is-Mobile-Viewer`, `CloudFront-Is-SmartTV-Viewer`, `CloudFront-Is-Tablet-Viewer`) and you configure your origin to return `Vary:User-Agent` to your distribution, your distribution returns `Vary:User-Agent` to the viewer.
 - If you configure your origin to include either `Accept-Encoding` or `Cookie` in the `Vary` header, Your distribution includes the values in the response to the viewer.
 - If you configure your distribution to forward an allow list of headers to your origin, and if you configure your origin to return the header names to your distribution in the `Vary` header (for example, `Vary:Accept-Charset, Accept-Language`), Your distribution returns the `Vary` header with those values to the viewer.
 - For information about how your distribution processes a value of `*` in the `Vary` header, see [Content negotiation](#).
 - If you configure your origin to include any other values in the `Vary` header, your distribution removes the values before returning the response to the viewer.
- **Via** – Your distribution sets the value to the following in the response to the viewer:

Via: *http-version alphanumeric-string*.cloudfront.net (CloudFront)

For example, if the client makes a request over HTTP/1.1, the value is something like the following:

Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)

Maximum file size

The maximum size of a response body that your distribution will return to the viewer is 20 GB. This includes chunked transfer responses that don't specify the `Content-Length` header value.

Origin unavailable

If your origin server is unavailable and your distribution gets a request for an object that is in the edge cache but that has expired (for example, because the period of time specified in the `Cache-Control max-age` directive has passed), your distribution either serves the expired version of the object or serves a custom error page.

In some cases, an object that is seldom requested is evicted and is no longer available in the edge cache. Your distribution can't serve an object that has been evicted.

Redirects

If you change the location of an object on your origin server, you can configure your web server to redirect requests to the new location. After you configure the redirect, the first time a viewer submits a request for the object, your distribution sends the request to the origin, and the origin responds with a redirect (for example, `302 Moved Temporarily`). Your distribution caches the redirect and returns it to the viewer. Your distribution does not follow the redirect.

You can configure your web server to redirect requests to one of the following locations:

- The new URL of the object on the origin server. When the viewer follows the redirect to the new URL, the viewer bypasses your distribution and goes straight to the origin. As a result, we recommend that you not redirect requests to the new URL of the object on the origin.
- The new distribution URL for the object. When the viewer submits the request that contains the new distribution URL, your distribution gets the object from the new location on your origin, caches it at the edge location, and returns the object to the viewer. Subsequent requests for the object will be served by the edge location. This avoids the latency and load associated with viewers requesting the object from the origin. However, every new request for the object will incur charges for two requests to your distribution.

Transfer encoding

Lightsail distributions support only the chunked value of the `Transfer-Encoding` header. If your origin returns `Transfer-Encoding: chunked`, your distribution returns the object to the

client as the object is received at the edge location, and caches the object in chunked format for subsequent requests.

If the viewer makes a `Range GET` request and the origin returns `Transfer-Encoding: chunked`, your distribution returns the entire object to the viewer instead of the requested range.

We recommend that you use chunked encoding if the content length of your response cannot be predetermined. For more information, see [Dropped TCP Connections](#).

Test your Lightsail distribution

In this guide, you'll learn how to test that your Amazon Lightsail distribution is caching and serving content from your origin. You should perform this test after you add your registered domain name to your distribution. For more information about distributions, see [Content delivery network distributions](#).

Test your distribution

Complete the following procedure to test your distribution. We use the Chrome web browser in this procedure; other browsers may use similar steps.

1. Open the Chrome web browser.
2. Open the **Chrome Menu** in the upper-right-hand corner of the browser window and select **More Tools > Developer Tools**.

You can also use the shortcut `Option + ⌘ + J` (on macOS), or `Shift + CTRL + J` (on Windows/Linux).

3. In the developer tools pane, choose the **Network** tab.
4. Browse to the domain of your distribution (e.g., `https://www.example.com`).

The **Network** tab of the Chrome developer tools should will populate with a list of objects from your website.

5. Choose a static object, such as an image file (.jpg, .png, .gif).
6. In the **Header** panel that appears, you should see that the `via` and `x-cache` headers both mention CloudFront. This confirms that your distribution is caching and serving content from your origin. your

The screenshot shows a browser window with the following content:

- Page Title:** user's Blog!
- Post Category:** UNCATEGORIZED
- Post Title:** Hello world!
- Post Metadata:** By user, February 19, 2020, 1 Comment
- Text:** Welcome to WordPress. This is your first post. Edit or delete it, then start writing!
- Image:** A hand-drawn orange robot character.

The browser's network tab is open, displaying a list of resources. The resource **sailbot.jpg** is selected. The response headers for this resource are:

- General:**
 - Request URL: https://robbox123.com/wp-content/uploads/2020/06/sa11bot.jpg
 - Request Method: GET
 - Status Code: 200
 - Remote Address: 99.84.71.178:443
 - Referrer Policy: no-referrer-when-downgrade
- Response Headers:**
 - accept-ranges: bytes
 - age: 8
 - cache-control: s-maxage=10
 - content-length: 48224
 - content-type: image/jpeg
 - date: Thu, 25 Jun 2020 12:11:46 GMT
 - etag: "bc60-5a8e774882d25"
 - last-modified: Thu, 25 Jun 2020 12:08:49 GMT
 - server: Apache
 - status: 200
 - via: 1.1 9b311162717b41c968f6f00426d88aaa.cloudfront.net (CloudFront)**
 - x-amz-cf-id: guY1UdZ6jaKfgBCNIw_EuYGD7ELa8zhPfqktKrF4GQaIKRokpCoM8A=
 - x-amz-cf-pop: SFO01-51
 - x-cache: Hit from cloudfront**
 - x-frame-options: SAMEORIGIN

Networking resources in Amazon Lightsail

Lightsail networking resources improve how users and outside services connect to your Lightsail instances.

Load balancers

You can create *load balancers* to add redundancy or to handle more traffic. For more information, see [Load balancers](#).

Static IPs

You can create *static IP addresses* to keep the same IP address every time you reboot your instance. For more information, see [Static IP addresses](#).

Regions and Availability Zones for Amazon Lightsail

When creating resources in Amazon Lightsail, create them in an AWS Region that is closest to your users. For example, if your blog traffic comes mostly from Switzerland, choose **Frankfurt** or **Paris**.

Note

DNS zones are global resources. They are created only in the US East (N. Virginia) (us-east-1) region, but they can reference any instance in any AWS Region.

Lightsail is available in the following AWS Regions:

- US East (Ohio) (us-east-2)
- US East (N. Virginia) (us-east-1)
- US West (Oregon) (us-west-2)
- Asia Pacific (Mumbai) (ap-south-1)
- Asia Pacific (Seoul) (ap-northeast-2)
- Asia Pacific (Singapore) (ap-southeast-1)
- Asia Pacific (Sydney) (ap-southeast-2)
- Asia Pacific (Tokyo) (ap-northeast-1)

- Canada (Central) (ca-central-1)
- EU (Frankfurt) (eu-central-1)
- EU (Ireland) (eu-west-1)
- EU (London) (eu-west-2)
- EU (Paris) (eu-west-3)
- EU (Stockholm) (eu-north-1)



SSH keys and Lightsail regions

In Lightsail, as soon as you create an instance in an AWS Region, we create a **Default** SSH key in that region. This default key can be used to connect to instances only in that specific region. To use the same key in all the regions where you have instances, create your own key pair and upload it to each of those regions. Or upload an existing key pair in those regions.

For more information, see [SSH key pairs](#).

Tips for working with Lightsail regions

Each AWS Region is designed to be completely isolated from other AWS Regions. This achieves the greatest possible fault tolerance and stability.

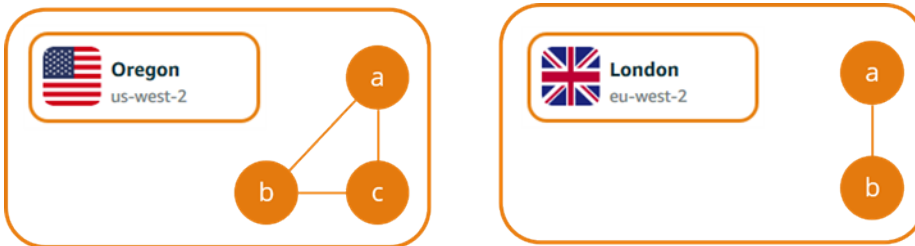
All communication between regions occurs across the public internet. Therefore, you should use the appropriate encryption methods to protect your data. Note that there is a charge for data transfer between regions. For more information, see [Amazon EC2 Pricing - Data Transfer](#).

When you work with a Lightsail instance using the AWS Command Line Interface (AWS CLI) or API operations, you must specify its regional endpoint. Use the `--region` option in your AWS CLI

command and specify `us-east-1` to return information about DNS zones and network resources. For more information about using the AWS CLI `--region` option, see [General Options](#) in the *AWS CLI Reference*.

Lightsail Availability Zones

Availability Zones are collections of data centers that run on physically distinct, independent infrastructure. Availability Zones are engineered to be highly reliable. Common points of failure such as generators and cooling equipment are not shared between Availability Zones. Availability Zones are also physically separate, so that even an extreme disaster such as a fire, tornado, or flood will affect only the single Availability Zone where it occurred.



Each AWS Region has multiple, isolated Availability Zones, which are indicated by a letter following the region name (`us-east-2a`). You can create Lightsail instances in only one Availability Zone at a time. You might not see all Availability Zones at the time you create your instance. If you don't see the list of Availability Zones at all, be sure that you have selected a region in the previous step.

Availability Zones and your Lightsail application

By launching your instances in separate Availability Zones, you can protect your applications from a failure in a single location.

To create an instance that is available in multiple Availability Zones, first [create a snapshot of your instance](#). Next, choose another Availability Zone when you [create a new instance from the snapshot you created](#).

For more information, see [AWS Regions and Availability Zones](#) in the *Amazon EC2 User Guide*.

Configure reverse DNS for an email server on your Amazon Lightsail instance

A reverse Domain Name System (DNS) lookup is used by email servers to track where a message originated from, and confirm that it's not spam or malicious. A reverse DNS lookup returns the

domain name of an IP address. This is in contrast to a forward DNS lookup, which returns the IP address of a domain.

For example, if a reverse DNS lookup of the IP address 192.168.1.2 returns the subdomain mail.example.com, and a forward DNS lookup of the subdomain mail.example.com returns the IP address 192.168.1.2, then reverse DNS for IP address 192.168.1.2 is forward-confirmed. To learn more, see [Forward-confirmed reverse DNS](#) on Wikipedia.

You can configure reverse DNS for your Amazon Lightsail instance by completing prerequisites, and then submitting a request to AWS Support to remove outbound messaging quotas. These steps are covered in the following sections.

Prerequisites

To configure reverse DNS, complete the following prerequisites in the order shown:

1. Create a Lightsail instance to be used as the email server. For more information, see [Create an instance](#).
2. Create a static IP to be used for the reverse DNS record, and attach it to your running instance. For more information, see [Create a static IP and attach it to an instance](#).

Important

You cannot use the default public IP, which is assigned to an instance when you first create it, for reverse DNS. This is because the default public IP for your instance changes when you stop and start your instance.

3. In your domain's DNS zone, add an alias record (A record) that points a subdomain, such as mail.example.com, to the static IP address of your running instance. This is the subdomain that is returned when a reverse DNS lookup of the static IP address is performed. For more information, see [Create a DNS zone to manage your domain's DNS records](#).

Note

We recommend that you transfer management of your domain's DNS records to Lightsail. This allows you to manage all of your resources, including your domain, in one place—the Lightsail console. For more information, see [Create a DNS zone to manage your domain's DNS records](#).

4. Allow time for changes to propagate through the internet's DNS. Then, you can continue submitting the request to AWS Support to configure reverse DNS.

Submit a request to AWS Support to configure reverse DNS

For security reasons, Lightsail limits outbound messages through port 25 by default. However, you can request AWS Support to remove this quota from your account and configure reverse DNS for your static IP.

To submit a request to AWS Support

1. Sign in to the [Lightsail console](#) as the AWS account root user.

Important

The request must be submitted using the AWS account root user. For more information about the AWS account root user, see [The AWS Account Root User](#).

2. Navigate to the [Request to Remove Email Sending Limitations](#) form, and enter the following required information:

Note

The form references Amazon Elastic Compute (EC2) resources, such as elastic IPs (EIPs) and EC2 instances. However, you can also use the form for your Lightsail resources, such as static IPs and Lightsail instances.

- **Email address** — Enter the email address where you can receive correspondence about your request. Your account email address is prepopulated in this text box.
 - **Use case description** — Enter the reason for requesting removal of the email quota.
 - **Elastic IP address** — Enter the static IP address that you attached to your instance in step 2 of the prerequisites earlier in this guide. You can enter up to two static IP addresses.
 - **Reverse DNS record for EIP** — Enter the subdomain that you defined in step 3 of the prerequisites earlier in this guide. This is the domain that is returned when the reverse DNS lookup is performed.
3. Choose **Submit** when done.

After your request is completed by AWS Support, your static IP address can be forward-confirmed with reverse DNS lookup.

If you later want to delete the static IP address from your Lightsail account, you must submit a request to AWS Support to remove the reverse DNS configuration. After the reverse DNS configuration is removed, you can delete the static IP address from your Lightsail account using the Lightsail console. For more information, see [Delete a static IP](#).

Set up Amazon VPC peering to work with AWS resources outside of Amazon Lightsail

With Lightsail, you can connect to AWS resources, such as an Amazon RDS database, through virtual private cloud (VPC) peering. A VPC is a virtual network dedicated to your AWS account. Everything you create inside Lightsail is inside a VPC, and you can connect your Lightsail VPC to an Amazon VPC.

Some AWS resources, such as Amazon S3, Amazon CloudFront, and Amazon DynamoDB don't require VPC peering to be enabled.

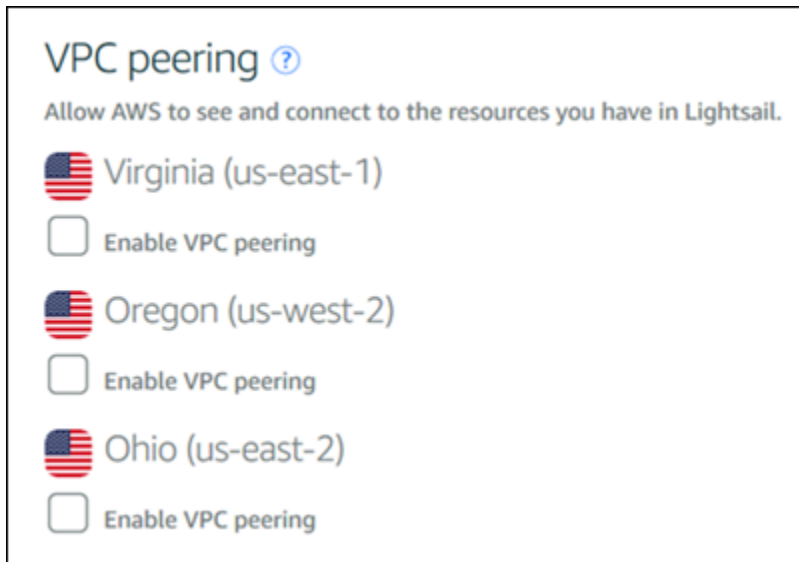
Note

To enable VPC peering in Lightsail, you need to have a default Amazon VPC. If you don't have a default Amazon VPC, you can create one. To learn more, see [Creating a Default VPC](#) in the *Amazon VPC User Guide*.

Since AWS Regions are isolated from one another, a VPC is also isolated in the region where you created it. You'll need to enable VPC peering in each region where you have Lightsail resources.

Once you have a default Amazon VPC, follow these instructions to peer your Lightsail VPC with your Amazon VPC.

1. In the [Lightsail console](#), choose **Account** on the top navigation menu.
2. Choose **Account** from the drop-down.
3. Choose the **Advanced** tab.
4. Choose **Enable VPC peering** under the AWS Region where you want to enable it.



If the peering connection fails, try to enable VPC peering again. If it doesn't work, contact [AWS Customer Support](#).

A peering connection is created in your AWS account if the peering request is successful. Go to the [Amazon VPC Dashboard](#) and choose **Peering Connections** in the navigation pane to view the peering connection that is created.

For more information about Amazon VPC, see [Your VPC and Subnets](#) in the *Amazon VPC User Guide*.

IP addresses in Amazon Lightsail

You can communicate with your Lightsail instance, and other Lightsail resources, using their IP addresses. For example, using the public IP address of your instance, you can check the network status of your instance (using PING), establish an SSH connection to your instance, and route traffic to your instance from a custom domain name. There are many more things you can do with the IP address of your Lightsail resources.

Lightsail instances, container services, and load balancers support both the IPv4 and IPv6 addressing protocols. These resources use the IPv4 addressing protocol by default; you can't disable this behavior. You can optionally enable IPv6 for your instances, container services, and load balancers.

In this guide, we cover what you need to know about IP addresses in Lightsail.

Contents

- [Private and public IPv4 addresses for instances](#)
- [Static IP addresses for instances](#)
- [IPv6 for instances, container services, CDN distributions, and load balancers](#)

Private and public IPv4 addresses for instances

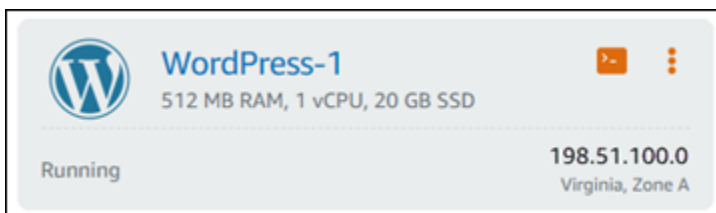
When you create a Lightsail instance, it is assigned a public and a private IPv4 address. The public IP address is accessible to the internet, while the private IP address is accessible only to resources in your Lightsail account in the same AWS Region.

Note

The private IP address of your instance can be accessible to other AWS resources in the same AWS Region, but outside of your Lightsail account, if you enable VPC peering. For more information, see [Set up Amazon VPC peering to work with AWS resources outside of Lightsail](#).

The IP addresses of your instance are displayed in the following areas of the Lightsail console:

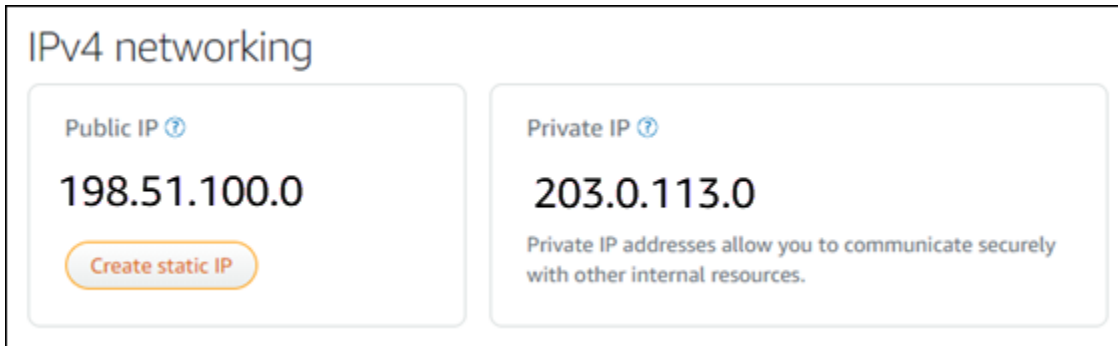
- The following example shows the public IP address of an instance on the Lightsail home page.



- The following example shows the public and private IP addresses of an instance in the header area of the instance management page.



- The following example shows the public and private IP addresses of an instance on the **Networking** tab of the instance management page.



Keep the following in mind when using the IPv4 addresses of your instances:

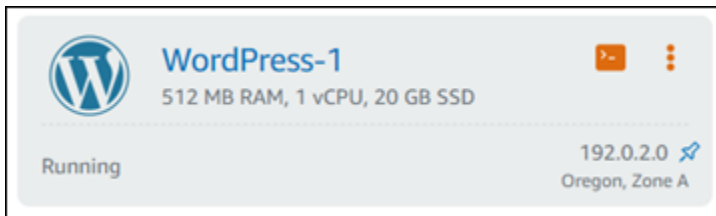
- The public IP address of your instance might change. Give your instance an IP address that never changes by attaching a static IP to it. For more information, see the [Static IP addresses for instances](#) section of this guide.
- Lightsail uses IPv4 addresses by default. However, You can optionally enable IPv6 for some Lightsail resources that were created before January 12, 2021. Resources created on or after January 12, 2021, have IPv6 enabled by default. For more information, see the [IPv6 for instances, container services, CDN distributions, and load balancers](#) section of this guide.
- Add rules to your instance firewall to control the traffic that is allowed to connect to it. For more information, see [Instance firewalls](#).

Static IPv4 addresses for instances

The default public IPv4 address that is assigned to your instance when you create it will change when you stop and start your instance. You can optionally create and attach a static IPv4 address to your instance. The static IPv4 address replaces the default public IPv4 address of your instance, and it stays the same when you stop and start your instance. You can attach one static IP to an instance. For more information, see [Create a static IP and attach it to an instance](#).

After you create a static IP, and attach it to your instance, it is displayed in the following areas of the Lightsail console:

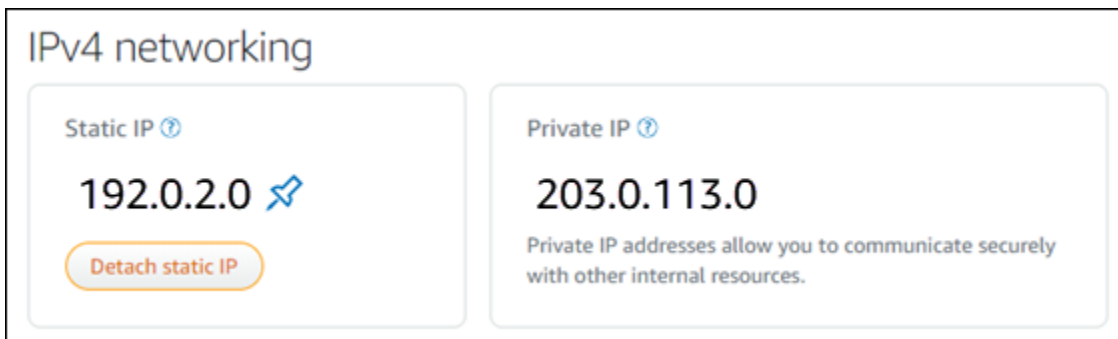
- The following example shows the static IP address of an instance on the Lightsail home page. The thumbtack icon signifies that the public IP address is static.



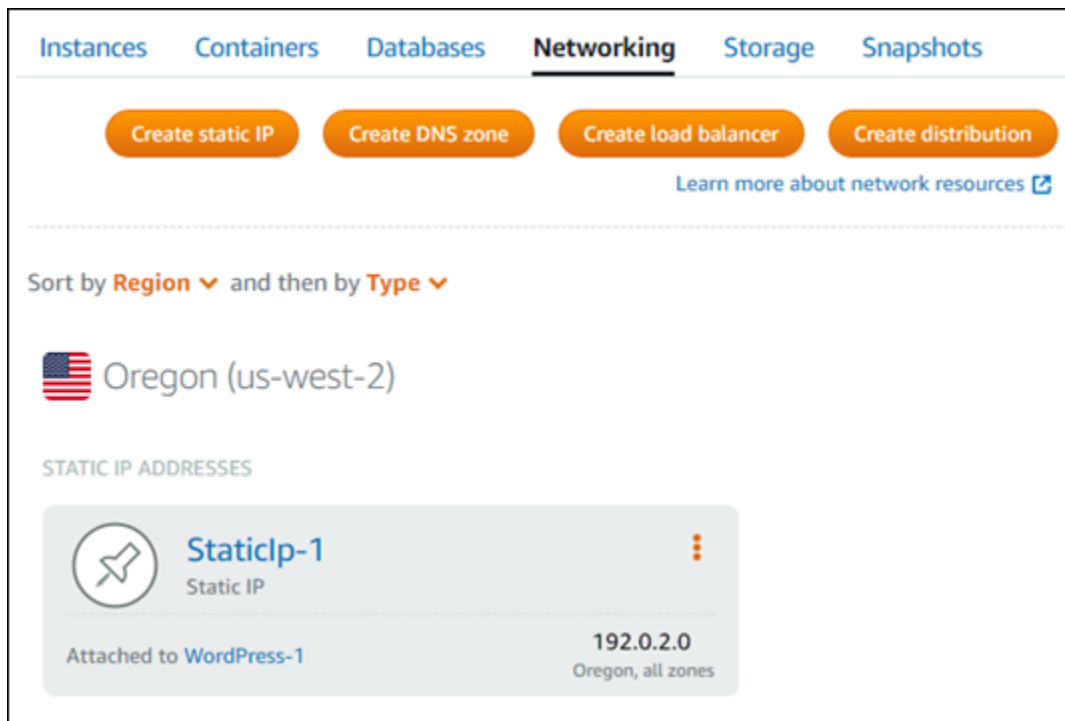
- The following example shows the static IP address of an instance in the header area of the instance management page. The thumbtack icon signifies that the public IP address is static.



- The following example shows the static IP address of an instance on the **Networking** tab of the instance management page. The default public IP address is no longer listed, and it has been replaced by the static IP address. The thumbtack icon signifies that the public IP address is static.



- You can view all of the static IPs that you've created by going to the Networking tab of the Lightsail home page as shown in the following example.



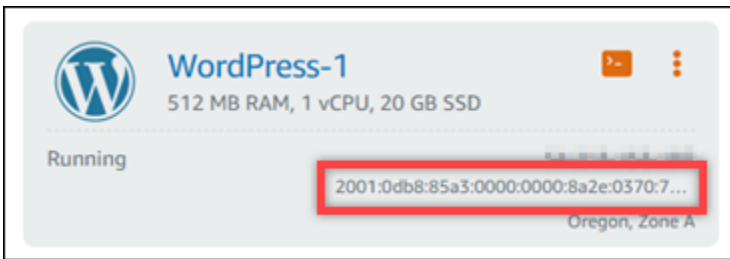
IPv6 for instances, container services, CDN distributions, and load balancers

IPv6 is enabled by default for Lightsail instances, container services, CDN distributions, and load balancers created on or after January 12, 2021. You can optionally enable IPv6 for those resources that were created before January 12, 2021. When you enable IPv6 for a specific resource, Lightsail automatically assigns an IPv6 address to that resource; you cannot choose or specify the IPv6 address yourself. For more information, see [Enable or disable IPv6](#).

You can also create an IPv6-only instance. An IPv6-only instance can communicate publicly over IPv6 only and does not have a public IPv4 address. For more information, see [IPv6-only instance plans in Lightsail](#)

Your instance's IPv6 address is displayed in the following areas of the Lightsail console:

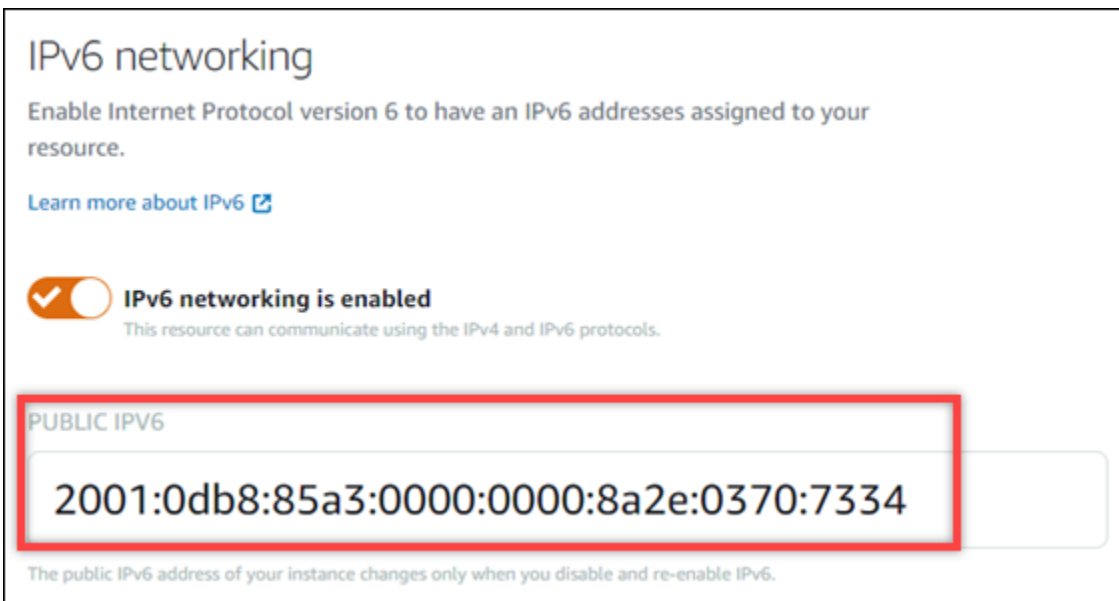
- The following example shows the IPv6 address of an instance on the Lightsail home page.



- The following example shows the IPv6 address of a resource in the header area of the resource's management page.



- The following example shows the IPv6 address of a resource on the Networking tab of the resource management page.



Keep the following in mind when you enable and use IPv6 for your resources:

- Your resources can communicate over IPv4 and IPv6 (in dual-stack mode) when you enable IPv6 for a resource, or over IPv4 only.

- When you enable IPv6 for a resource, Lightsail automatically assigns an IPv6 address to that resource; you cannot choose or specify the IPv6 address yourself. When you enable IPv6 for a resource, it begins accepting network traffic over the IPv6 protocol.
- The IPv6 address for an instance persists when you stop and start your instance. It's released only when you delete your instance, or disable IPv6 for your instance. You cannot get the IPv6 address back after you perform either of those actions.
- All IPv6 addresses that are assigned to your instances are public and reachable over the internet. There are no private IPv6 addresses that are assigned to your instances.
- IPv4 and IPv6 addresses for instances are independent of each other; you must configure instance firewall rules separately for IPv4 and IPv6. For more information, see [Instance firewalls](#).
- Not all instance blueprints available in Lightsail are automatically configured for IPv6 when IPv6 is enabled. Instances that use the following blueprints require additional configuration steps after you enable IPv6 for them:
 - **cPanel** – For more information, see [Configure IPv6 for cPanel instances](#).
 - **Debian 8** – For more information, see [Configure IPv6 for Debian 8 instances](#).
 - **GitLab** – For more information, see [Configure IPv6 for GitLab instances](#).
 - **Nginx** – For more information, see [Configure IPv6 for Nginx instances](#).
 - **Plesk** – For more information, see [Configure IPv6 for Plesk instances](#).
 - **Ubuntu 16** – For more information, see [Configure IPv6 for Ubuntu 16 instances](#).

Note

PrestaShop does not currently support IPv6 addresses. You can enable IPv6 for the instance, but the PrestaShop software will not respond to requests over the IPv6 network.

Static IP addresses in Amazon Lightsail

A static IP is a fixed, public IP address that you can assign and reassign to an instance or other resource. If you haven't set up a static IP address, each time you stop or restart your instance, Lightsail assigns a new public IP address.

Important

If you stop or restart your instance without first creating a static IP address and attaching it to your instance, you lose your IP address when your instance restarts. You should create a static IP address and attach it to your instance to ensure that your instance always has the same public IP address. For more information, see [Create a static IP address](#).

Contents

- [Create a static IP address and attach it to a Lightsail instance](#)
- [Delete a static IP address in Lightsail](#)

Create a static IP address and attach it to a Lightsail instance

The default dynamic public IP address attached to your Amazon Lightsail instance changes every time you stop and restart the instance. Create a static IP address and attach it to your instance to keep the public IP address from changing. Later, when you point a registered domain name to your instance, you don't have to update your domain's DNS records every time you stop and restart your instance. You can attach one static IP to an instance. For more information, see [Static IP addresses](#).

Prerequisites

You need at least one dual-stack instance running in Lightsail. To create one, see [Create an instance](#).

Create and assign a Static IP address to an instance

Follow these steps to create a new static IP address and attach it to an instance in Lightsail.

1. Sign in to the Lightsail console at <https://lightsail.aws.amazon.com/>.
2. On the Lightsail home page, choose **Networking**.
3. Choose **Create static IP**.
4. Select the AWS Region where you want to create your static IP.

Note

Static IP addresses can only be attached to instances in the same Region.

5. Choose the Lightsail resource to which you want to attach the static IP.
6. Enter a name for your static IP.

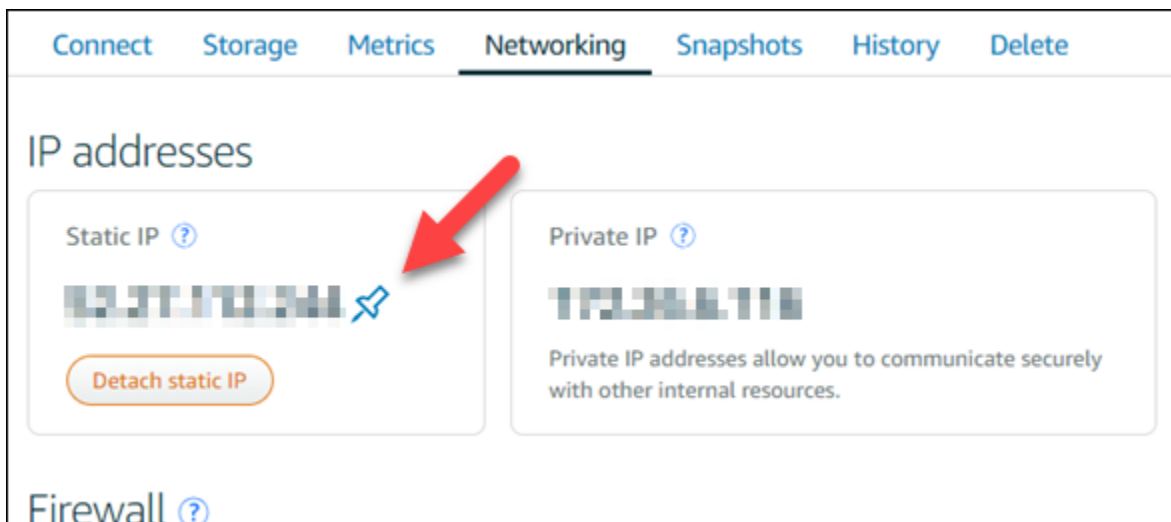
Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
7. Choose **Create**.

Now when you go to the home page, you see a static IP address that you can manage.



Also, on the **Networking** tab of your instance's management page, you'll see a blue pushpin next to your public IP address. This indicates that the IP address is now static.



For more information, see [Public and private IP addresses](#).

Delete a static IP address in Lightsail

You can create up to five static IPs per AWS Region in your Amazon Lightsail account. If you delete an instance that has a static IP address attached to it, the static IP address remains in your account. If you no longer need the static IP address, you can delete it using the Lightsail console or the AWS Command Line Interface (AWS CLI). In this guide, we show you how to delete a static IP address from your Lightsail account. For more information about static IPs, see [IP addresses](#).

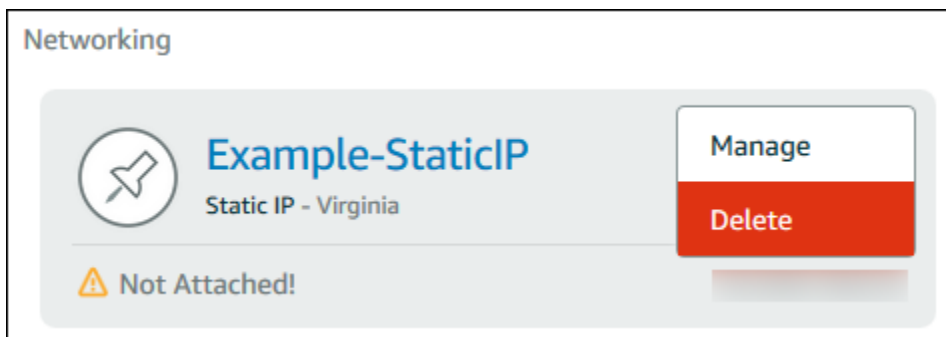
⚠ Important

Deleting a static IP will completely remove the static IP from your Lightsail account. Resources that use that static IP, such as instances, will be impacted. You will not be able to get the static IP back after you delete it.

Delete a static IP using the Lightsail console

Complete the following procedure to delete a static IP using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose **Networking**.
3. On the **Networking** page choose the vertical ellipsis (⋮) icon next to the static IP address that you want to delete, and then choose **Delete**.



Delete a static IP using the AWS CLI

Complete the following procedure to delete a static IP using the AWS CLI. The command to delete a static IP from your Lightsail account is [release-static-ip](#). When you create a static IP, you're actually *allocating* it. So, instead of deleting the static IP, you're actually *releasing* it.

Prerequisites

First, if you haven't already, you need to install the AWS CLI. To learn more, see [Installing the AWS Command Line Interface](#). Be sure [you configure the AWS CLI](#).

You'll need the name of your static IP to release it. You can get that by using the `get-static-ips` AWS CLI command.

1. Type the following command:

```
aws lightsail get-static-ips
```

You should see output similar to the following.

```
{
  "staticIps": [
    {
      "name": "Example-StaticIP",
      "resourceType": "StaticIp",
      "attachedTo": "MyInstance",
      "arn": "arn:aws:lightsail:us-east-2:123456789101:StaticIp/5282f35e-
c720-4e5a-1234-12345EXAMPLE",
      "isAttached": true,
      "ipAddress": "192.0.2.0",
      "createdAt": 1489750629.026,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-east-2"
      }
    },
    {
      "name": "my-other-static-ip",
      "resourceType": "StaticIp",
      "arn": "arn:aws:lightsail:us-east-2:123456789101:StaticIp/
f5885e14-8984-49e5-1234-12345EXAMPLE",
      "isAttached": false,
      "ipAddress": "192.0.2.2",
      "createdAt": 1483653597.815,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-east-2"
      }
    }
  ]
}
```

```
}
```

2. Select the **name** value of the static IP you wish to release and make a note of it so you can use it in the next step.

For example, you can copy the value to the clipboard.

3. Type the following command.

```
aws lightsail release-static-ip --static-ip-name StaticIpName
```

In the command, replace *StaticIpName* with the name of your static IP.

If successful, you should see output similar to the following.

```
{
  "operations": [
    {
      "status": "Succeeded",
      "resourceType": "StaticIp",
      "isTerminal": true,
      "statusChangedAt": 1489860944.19,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-east-2"
      },
      "operationType": "ReleaseStaticIp",
      "resourceName": "Example-StaticIP",
      "id": "92a2f0d2-eef2-4e6f-1234-12345EXAMPLE",
      "createdAt": 1489860944.19
    }
  ]
}
```

Enable and disable IPv6 in Amazon Lightsail

IPv6 is enabled by default for Lightsail instances, container services, CDN distributions, and load balancers created on or after January 12, 2021. You can optionally enable IPv6 for those resources that were created before January 12, 2021. In this guide, we show you how to enable or disable IPv6. For more information about IPv6, see [IP addresses](#).

Contents

- [Considerations for using IPv6](#)
- [Enable IPv6](#)
- [Disable IPv6](#)

IPv6 considerations

IPv6 became available in Lightsail on January 12, 2021; therefore, you might need to manually enable or disable IPv6 for some of your resources according to the following guidelines:

- Instances, CDN distributions, and load balancers created *before* January 12 have IPv6 disabled until you enable it. However, instances, CDN distributions, and load balancers created *after* January 12 have IPv6 enabled when they are created.
- Container services created *before* or *after* January 12 have IPv6 enabled.
- IPv6 can be manually enabled or disabled for instances, CDN distributions, and load balancers at any time. It cannot be disabled for container services.

Keep the following in mind when you enable and use IPv6:

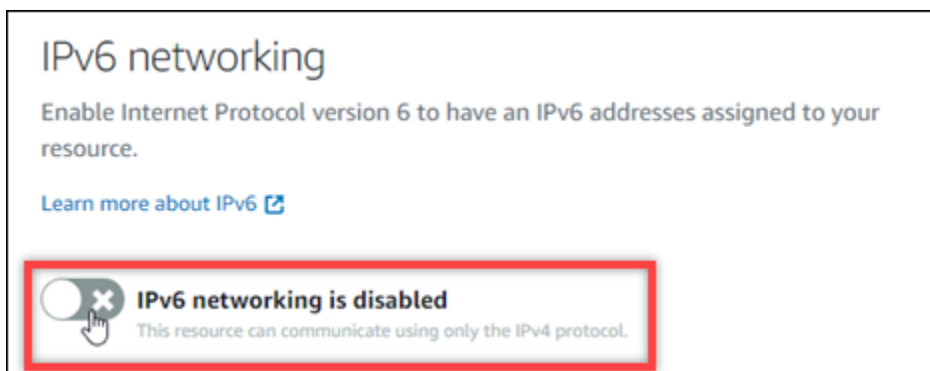
- Your resources can communicate over IPv4 only, or over IPv4 and IPv6 (in dual-stack mode) when you enable IPv6 for a resource.
- When you enable IPv6 for an instance, Lightsail automatically assigns an IPv6 address to that instance; you cannot choose or specify the IPv6 address yourself. When you enable IPv6 for a container service, CDN distribution, or load balancer, that resource will begin accepting internet traffic over IPv6.
- The IPv6 address for an instance persists when you stop and start your instance. It's released only when you delete your instance, or disable IPv6 for your instance. You cannot get the IPv6 address back after you perform either of those actions.
- All IPv6 addresses that are assigned to your instances are public and reachable over the internet. There are no private IPv6 addresses that are assigned to your instances.
- IPv4 and IPv6 addresses for instances are independent of each other; you must configure instance firewall rules separately for IPv4 and IPv6. For more information, see [Instance firewalls](#).

- Not all instance blueprints available in Lightsail are automatically configured for IPv6 when IPv6 is enabled. Instances that use the following blueprints require additional configuration steps after you enable IPv6 for them:
 - **cPanel** – For more information, see [Configure IPv6 for cPanel instances](#).
 - **Debian 8** – For more information, see [Configure IPv6 for Debian 8 instances](#).
 - **GitLab** – For more information, see [Configure IPv6 for GitLab instances](#).
 - **Nginx** – For more information, see [Configure IPv6 for Nginx instances](#).
 - **Plesk** – For more information, see [Configure IPv6 for Plesk instances](#).
 - **Ubuntu 16** – For more information, see [Configure IPv6 for Ubuntu 16 instances](#).

Enable IPv6

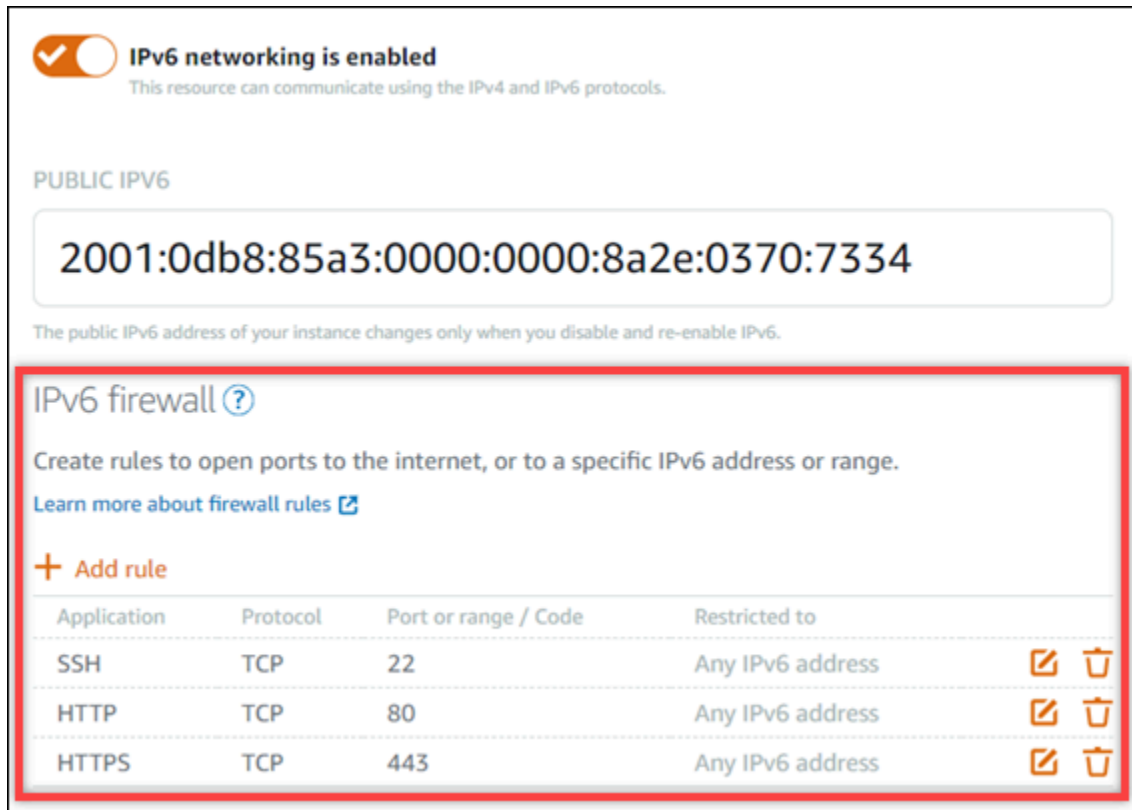
Complete the following procedure to enable IPv6 for instances, CDN distributions, and load balancers.

1. Sign in to the [Lightsail console](#).
2. Complete one of the following steps depending on the resource for which you want to enable IPv6:
 - To enable IPv6 for an instance, choose the **Instances** tab on the Lightsail home page, and then choose the name of the instance for which you want to enable IPv6.
 - To enable IPv6 for a CDN distribution or a load balancer, choose the **Networking** tab on the Lightsail home page, and then choose the name of the CDN distribution or load balancer for which you want to enable IPv6.
3. Choose the **Networking** tab in the resource's management page.
4. In the **IPv6 Networking** section of the page, choose the toggle to enable IPv6 for the resource.



Be aware of the following items after you enable IPv6 for a resource:

- If you enable IPv6 for a CDN distribution or load balancer, then that resource begins accepting IPv6 traffic. If you enable IPv6 for an instance, then an IPv6 address is assigned to it, and the IPv6 firewall becomes available, as shown in the following example.



IPv6 networking is enabled
This resource can communicate using the IPv4 and IPv6 protocols.

PUBLIC IPV6

2001:0db8:85a3:0000:0000:8a2e:0370:7334

The public IPv6 address of your instance changes only when you disable and re-enable IPv6.

IPv6 firewall ⓘ

Create rules to open ports to the internet, or to a specific IPv6 address or range.
[Learn more about firewall rules](#)

+ Add rule

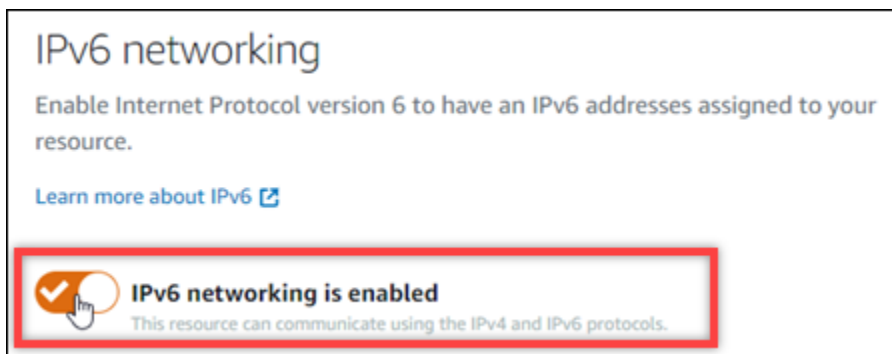
Application	Protocol	Port or range / Code	Restricted to		
SSH	TCP	22	Any IPv6 address	🗑️	🗑️
HTTP	TCP	80	Any IPv6 address	🗑️	🗑️
HTTPS	TCP	443	Any IPv6 address	🗑️	🗑️

- Instances that use the following blueprints require additional steps after enabling IPv6 to ensure the instance becomes aware of its new IPv6 address:
 - **cPanel** – For more information, see [Configure IPv6 for cPanel instances](#).
 - **Debian 8** – For more information, see [Configure IPv6 for Debian 8 instances](#).
 - **GitLab** – For more information, see [Configure IPv6 for GitLab instances](#).
 - **Nginx** – For more information, see [Configure IPv6 for Nginx instances](#).
 - **Plesk** – For more information, see [Configure IPv6 for Plesk instances](#).
 - **Ubuntu 16** – For more information, see [Configure IPv6 for Ubuntu 16 instances](#).
- If you have a registered domain name directing traffic to you instance, container service, CDN distribution, or load balancer, then make sure to create an IPv6 address record (AAAA) in the DNS of your domain to route IPv6 traffic to your resource.

Disable IPv6

Complete the following procedure to disable IPv6 for instances, CDN distributions, and load balancers.

1. Sign in to the [Lightsail console](#).
2. Complete one of the following steps depending on the resource for which you want to disable IPv6:
 - To disable IPv6 for an instance, choose the **Instances** tab on the Lightsail home page, and then choose the name of the instance for which you want to disable IPv6.
 - To disable IPv6 for a CDN distribution or a load balancer, choose the **Networking** tab on the Lightsail home page, and then choose the name of the CDN distribution or load balancer for which you want to disable IPv6.
3. Choose the **Networking** tab in the resource's management page.
4. In the **IPv6 Networking** section of the page, choose the toggle to disable IPv6 for the resource.



SSL/TLS certificates in Amazon Lightsail

Amazon Lightsail uses SSL/TLS certificates to validate custom (registered) domains that you can use with Lightsail load balancers, content delivery network (CDN) distributions, and container services. After a validated certificate is attached to one of those Lightsail resources, the traffic that is routed to that resource through the domain is encrypted using Hypertext Transfer Protocol Secure (HTTPS).

You can create Transport Layer Security (TLS) certificates in Amazon Lightsail to enable encrypted web traffic for custom (registered) domains that you want to use with your Lightsail load balancers content delivery network distributions, and container services. TLS is an updated, more secure

version of Secure Socket Layer (SSL). Throughout the Lightsail documentation and console, you'll see us refer to it as **SSL/TLS**.

Note

The Lightsail certificates that you can attach to load balancers, CDN distributions, and container services are issued by the AWS Certificate Manager (ACM) service. Starting October 11, 2022, any public certificate obtained through Lightsail for your load balancers, CDN distributions, and container services will be issued from one of the multiple intermediate certificate authorities (ICAs) or subordinate CAs that ACM manages. For more information, see [Amazon introduces dynamic intermediate certificate authorities](#) in the *AWS Security Blog*.

Why use HTTPS?

First and foremost is security. HTTPS offers an extra layer of security because it uses TLS to move data. HTTPS encryption is confidential between the web server and the client's browser, because they are the only two entities who can decrypt the traffic. HTTPS connections are also more secure because the data a client exchanges with the server can't be modified by another party.

Aside from security benefits mentioned above, there are other reasons to use HTTPS in addition to HTTP. For example, in 2014 Google began ranking secure websites higher in search results. In other words, a site that uses HTTPS ranks closer to the top of search results compared to a site that only uses HTTP (all other things being equal).

[Learn more about HTTPS as a ranking signal](#)

Process overview

The process to use a Lightsail certificate is simple. It involves the following steps:

1. Create your Lightsail resource that can use a Lightsail certificate, such as a load balancer, CDN distribution, or container service.
2. Create a certificate for your domain using Lightsail.
3. Validate the certificate by adding a canonical name (CNAME) record to the DNS of your domain
4. Attach the validated certificate to your Lightsail resource.

5. Modify the DNS of your domain to route traffic to your Lightsail resource.



After the certificate is attached to the resource, the traffic that is routed to that resource through the domain is encrypted using HTTPS.

Use SSL/TLS certificates with your distribution or container service

HTTPS is required on Lightsail distributions and container services. When you create either of those resources, HTTPS is enabled by default for the resource's default domain (e.g., `https://123456abcdef.cloudfront.net/` for a distribution or `https://container-service-1.123456abcdef.us-west-2.cs.amazonlightsail.com/` for a container service). If you want to use your registered domain name (e.g., `example.com`) with your distribution or container service, you must create a Lightsail SSL/TLS certificate, validate it with your domain name, and enable custom domains on your resource. Enabling custom domains on your distribution or container service also attaches your domain's validated certificate to your resource.

You can get started with enabling custom domains and HTTPS on your distribution by following these links.

- [Create SSL/TLS certificates for your distribution](#)
- [Validate SSL/TLS certificates for your distribution](#)
- [View SSL/TLS certificates for your distribution](#)
- [Enable custom domains for your distribution](#)
- [Point your domain to a distribution](#)

For more information about distributions, see [Content delivery network distributions](#).

You can get started with enabling custom domains and HTTPS on your container service by following these links.

- [Create container service SSL/TLS certificates](#)
- [Validate container service SSL/TLS certificates](#)
- [Enable and manage custom domains](#)

For more information about container services, see [Container services](#).

Use SSL/TLS certificates with your load balancer

When you create a Lightsail load balancer, port 80 is open by default to handling regular HTTP traffic. To enable HTTPS traffic over port 443, you must create an SSL/TLS certificate, validate it with your domain name, and attach it to your load balancer.

You can create up to two SSL/TLS certificates per load balancer. Only one certificate can be in use at a time per load balancer. If you delete a valid, in-use certificate from your load balancer, your load balancer is no longer be able to handle HTTPS traffic for the specified domain until you attach another valid certificate.

You can get started with enabling HTTPS on your load balancer by following these links.

- [Create a load balancer and attach instances to it](#)
- [Create an SSL/TLS certificate](#)
- [Verify domain ownership](#)
- [Attach your validated certificate to enable HTTPS](#)

For more information about load balancers, see [Load balancers](#).

Lightsail container service SSL/TLS certificates

You can create Amazon Lightsail TLS/SSL certificates for your Lightsail container service. When you create a certificate, you specify the primary and alternate domain names for the certificate. When you enable custom domains for your container service, and choose the certificate, you can choose up to four domains from the certificate that will be added as the custom domains of your container service. After you update the DNS record of your domains to direct traffic to your container service, your service accepts the traffic and serves your content using HTTPS. There is a quota for the number of certificates that you can create. For more information, see [Lightsail service quotas](#).

For more information about SSL/TLS certificates, see [Container service certificates](#).

Prerequisites

Before you get started, you need to create a Lightsail container service. For more information, see [Create a container services](#) and [Container services](#).

Create an SSL/TLS certificate for your container service

Complete the following procedure to create an SSL/TLS certificate for your container service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which want to create a certificate.
4. Choose the **Custom domains** tab on your container service management page.
5. Scroll down to the **Attached certificates** section of the page.

All of your certificates are listed under the Attached certificates section of the page, including certificates created for other Lightsail resources, and certificates that are in use and not in use.

6. Choose **Create certificate**.
7. Enter a unique name in the **Certificate name** text box to identify your certificate. Then, choose **Continue**.
8. Enter the primary domain name (e.g., `example.com`) that you want to use with the certificate into the **Specify up to 10 domains or subdomains** field.
9. (Optional) Enter another domain name (e.g., `www.example.com`) into the **Specify up to 10 domains or subdomains** field.

You can add up to nine alternate domains to your certificate. You can use up to four of your certificate's domains with your container service after you enable custom domains and select the certificate for your service.

10. Choose **Create certificate**.

Your certificate request is submitted, and the status of your new certificate is changed to **Attempting to validate your certificate**. During this time, Lightsail attempts to add the certificate's validation record to the DNS of the primary domain. After a while, the status will change to **Valid**.

If automatic validation fails you will be required to validate the certificate with your domains before you can use it with your container service. For more information, see [Validate container service SSL/TLS certificates](#).

Topics

- [Validate Lightsail container service SSL/TLS certificates](#)
- [View Lightsail container service SSL/TLS certificates](#)

Validate Lightsail container service SSL/TLS certificates

An Amazon Lightsail SSL/TLS certificate must be validated after it's created, and before you can use it with your Lightsail container service. After your certificate request is submitted, the status of your new certificate is changed to **Attempting to validate your certificate**. During this time, Lightsail attempts to add the certificate's validation record to the DNS of the domain names that you specified for the certificate. After a while, the status will change to **Valid**, or **Validation timed out**.

If automatic validation fails you must verify that you control all the domain names that you specified for the certificate when you created it. You do this by adding canonical name (CNAME) records to the DNS zone of each of the domains specified on the certificate. The records that you need to add are listed in the **Validation details** section of the certificate.

In this guide, we provide you with the procedure to manually validate your certificate using a Lightsail DNS zone. The procedure to validate your certificate using a different DNS hosting provider, like Domain.com or GoDaddy, might be similar. For more information about Lightsail DNS zones, see [DNS](#).

For more information about SSL/TLS certificates, see [SSL/TLS certificates](#).

Prerequisite

Before you get started, you need to create an SSL/TLS certificate for your container service. For more information, see [Create SSL/TLS certificates for your container services](#).

Get the CNAME record values to validate your certificate

Complete the following procedure to get the CNAME records that you must add to your domains to validate the certificate.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which want to create a certificate.
4. Choose the **Custom domains** tab on your container service management page.
5. Scroll down to the **Attached certificates** section of the page.

All of your certificates are listed under the **Attached certificates** section of the page, including certificates created for other Lightsail resources, and certificates that are pending validation.

6. Find the certificate that you want to validate, expand **Validation details**, and make note of the **Name** and **Value** of the CNAME records that you must add for each domain listed.

You must add these records exactly as listed. We recommend that you copy and paste these values into a text file that you can refer to later. For more information, see the following [Add the CNAME records to your domain's DNS zone](#) section of this guide.

Add the CNAME records to your domain's DNS zone

Complete the following procedure to add CNAME records to your domain's DNS zone.

1. On the Lightsail home page, choose the **Domains & DNS** tab.
2. Under the **DNS zones** section of the page, choose the domain name to which you want to add the CNAME records to validate your certificate.
3. Choose the **DNS records** tab.
4. Choose **Add record** in the DNS records management page.
5. Choose **CNAME** in the **Record type** drop-down.
6. In the **Record name** text box, enter the **Name** value of the CNAME record that you got from your certificate.

The Lightsail console pre-populates the apex portion of your domain. For example, if you want to add the `www.example.com` subdomain, then you only have to enter `www` into the text box, and Lightsail adds the `.example.com` portion for you when you save the record.

7. In the **Route traffic to** text box, enter the **Value** portion of the CNAME record that you got from your certificate.
8. Confirm that the values you entered are exactly as they were listed on the certificate that you want to validate.

9. Choose the save icon to save the record to your DNS zone.

Repeat these steps to add additional CNAME records for domains on your certificate that need to be validated. Allow time for changes to propagate through the internet's DNS. After a few minutes, you should see if the status of your certificate has changed to **Valid**. For more information, see the following [View the status of your certificate](#) section of this guide.

View the status of your certificate

Complete the following procedure to view the status of your SSL/TLS certificate.

1. On the Lightsail home page, choose the **Containers** tab.
2. Choose the name of the container service for which you want to view a certificate's status.
3. Choose the **Custom domains** tab on your container service management page.
4. Scroll down to the **Attached certificates** section of the page.

All of your certificates are listed under the **Attached certificates** section of the page, including certificates with **Pending** validation and **Valid** statuses.

Note

If you left the **Custom domains** page open while validating your certificates, you might have to refresh to see the updated status of your certificates.

A **Valid** status confirms that you successfully validated your certificate with the CNAME records that you added to your domains. Choose **Details** to view your certificate's important dates, encryption details, identification, and validation records. Your certificates are valid for 13 months from the date on which you validated them, after which time Lightsail attempts to automatically re-validate them. Don't delete the CNAME records that you added to your domain because they are required when your certificate is re-validated on the **Valid until** date listed.

After you validate your SSL/TLS certificate, you should enable custom domains for your container service to use the domain names of your certificate on your service. For more information, see [Enable and manage custom domains for your container services](#).

View Lightsail container service SSL/TLS certificates

You can view the Amazon Lightsail SSL/TLS certificates that you created for your Lightsail container service. You do this by accessing the management page of any container service in the Lightsail console.

For more information about SSL/TLS certificates, see [SSL/TLS certificates](#).

Prerequisites

Before you get started, you need to create a Lightsail container service. For more information, see [Creating Amazon Lightsail container services](#) and [Container services](#).

You also should have created an SSL/TLS certificate for your container service. For more information, see [Create container service SSL/TLS certificates](#).

View your container service SSL/TLS certificates

Complete the following procedure to view your container service SSL/TLS certificates.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of a container service.

You can view all of your certificates regardless of the container service you choose.

4. Choose the **Custom domains** tab on your container service management page.
5. Scroll down to the **Attached certificates** section of the page.

All of your certificates are listed under the **Attached certificates** section of the page. Choose **Details** to view your certificate's important dates, encryption details, identification, and domains. Choose **Validation details** to view your certificate's validation records. Your certificates are valid for 13 months from the date you created them, after which time Lightsail attempts to automatically revalidate them. Don't delete the CNAME records that you added to your domain because they are required when your certificate is re-validated on the **Valid until** date listed.

After you have a valid SSL/TLS certificate to use with your container service, you should enable custom domains so that you can use the domain names of the certificate on your service. For more information, see [Enable and manage custom domains](#).

Lightsail distribution SSL/TLS certificates

You can create Amazon Lightsail TLS/SSL certificates for your Lightsail distributions. When you create a certificate, you specify the primary and alternate domain names for the certificate. When you enable custom domains for your distribution, and choose the certificate, those domains are added as the custom domains of your distribution. After you update the DNS record of your domains to point to your distribution, your distribution accepts the traffic and serves your content using HTTPS. There is a quota for the number of certificates that you can create. For more information, see [Lightsail service quotas](#).

For more information about SSL/TLS certificates, see [SSL/TLS certificates](#).

Important

The domain names you specify when creating an SSL/TLS certificate for your distribution cannot be in use by another distribution across all Amazon Web Services (AWS) accounts, including distributions on the Amazon CloudFront service. You will be able to create the certificate for the domains, but you will not be able to use the certificate with your distribution.

Prerequisite

Before you get started, you need to create a Lightsail distribution. For more information, see [Create a distribution](#) and [Content delivery network distributions](#).

Create an SSL/TLS certificate for your distribution

Complete the following procedure to create an SSL/TLS certificate for your distribution.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which want to create a certificate.
4. Choose the **Custom domains** tab on your distribution's management page.
5. Scroll down to the **Attached certificates** section of the page.

All of your distribution certificates are listed under the **Attached certificates** section of the page, including certificates created for other distributions, and certificates that are in use and not in use.

6. Choose **Create certificate**.
7. Enter a unique name in the **Certificate name** text box to identify your certificate. Then, choose **Continue**.
8. Enter the primary domain name (e.g., `example.com`) that you want to use with the certificate into the **Specify up to 10 domains or subdomains** field.
9. (Optional) Enter alternate domain names (e.g., `www.example.com`) into the remaining **Specify up to 10 domains or subdomains** fields.

You can add up to nine alternate domains to your certificate. You will be able to use all of your certificate's domains with your distribution after you enable custom domains and select the certificate for your distribution.

10. Choose **Create**.

Your certificate request is submitted, and the status of your new certificate is changed to **Attempting to validate your certificate**. During this time, Lightsail attempts to add the certificate's validation record to the DNS of the primary domain. After a while, the status will change to **Valid**.

If automatic validation fails, you will be required to validate the certificate with your domains before you can use it with your distribution. For more information, see [Validate SSL/TLS certificates for your distribution](#).

Topics

- [View SSL/TLS certificates for your Lightsail distribution](#)
- [Validate SSL/TLS certificates for your Lightsail distribution](#)
- [Configure the minimum TLS protocol version for your Lightsail distribution certificate](#)
- [Delete SSL/TLS certificates for your Lightsail distribution](#)

View SSL/TLS certificates for your Lightsail distribution

You can view the Amazon Lightsail SSL/TLS certificates that you created for your Lightsail distributions. You do this by accessing the management page of any distribution in the Lightsail console.

For more information about SSL/TLS certificates, see [SSL/TLS certificates](#).

Prerequisites

Before you get started, you need to create a Lightsail distribution. For more information, see [Create a distribution](#) and [Content delivery network distributions](#).

You also should have created an SSL/TLS certificate for your distribution. For more information, see [Create SSL/TLS certificates for your distribution](#).

View your distribution SSL/TLS certificates

Complete the following procedure to view your distribution SSL/TLS certificates.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of a distribution.

You can view all of your certificates regardless of the distribution you choose.

4. Choose the **Custom domains** tab on your distribution's management page.
5. Scroll down to the **Attached certificates** section of the page.

All of your distribution certificates are listed under the **Attached certificates** section of the page. Expand **Validation details** to view your certificate's important dates, encryption details, identification, and validation records. Your certificates are valid for 13 months from the date you created them, after which time Lightsail attempts to automatically revalidate them. Don't delete the CNAME records that you added to your domain because they are required when your certificate is re-validated on the **Valid until** date listed.

After you have a valid SSL/TLS certificate to use with your distribution, you should enable custom domains so that you can use the domain names of the certificate on your distribution. For more information, see [Enable custom domains for your distribution](#).

Validate SSL/TLS certificates for your Lightsail distribution

An Amazon Lightsail SSL/TLS certificate must be validated after it's created, and before you can use it with your Lightsail distribution. After your certificate request is submitted, the status of your new certificate is changed to **Attempting to validate your certificate**. During this time, Lightsail attempts to add the certificate's validation record to the DNS of the domain names that you specified for the certificate. After a while, the status will change to **Valid**, or **Validation timed out**.

If automatic validation fails you must verify that you control all the domain names that you specified for the certificate when you created it. You do this by adding canonical name (CNAME) records to the DNS zone of each of the domains specified on the certificate. The records that you need to add are listed in the **Validation details** section of the certificate.

In this guide, we provide you with the procedure to manually validate your certificate using a Lightsail DNS zone. The procedure to validate your certificate using a different DNS hosting provider, like Domain.com or GoDaddy, may be similar. For more information about Lightsail DNS zones, see [DNS](#).

For more information about SSL/TLS certificates, see [SSL/TLS certificates](#).

Contents

- [Prerequisite](#)
- [Get the CNAME record values to validate your certificate](#)
- [Add the CNAME records to your domain's DNS zone](#)
- [View the status of your distribution certificate](#)

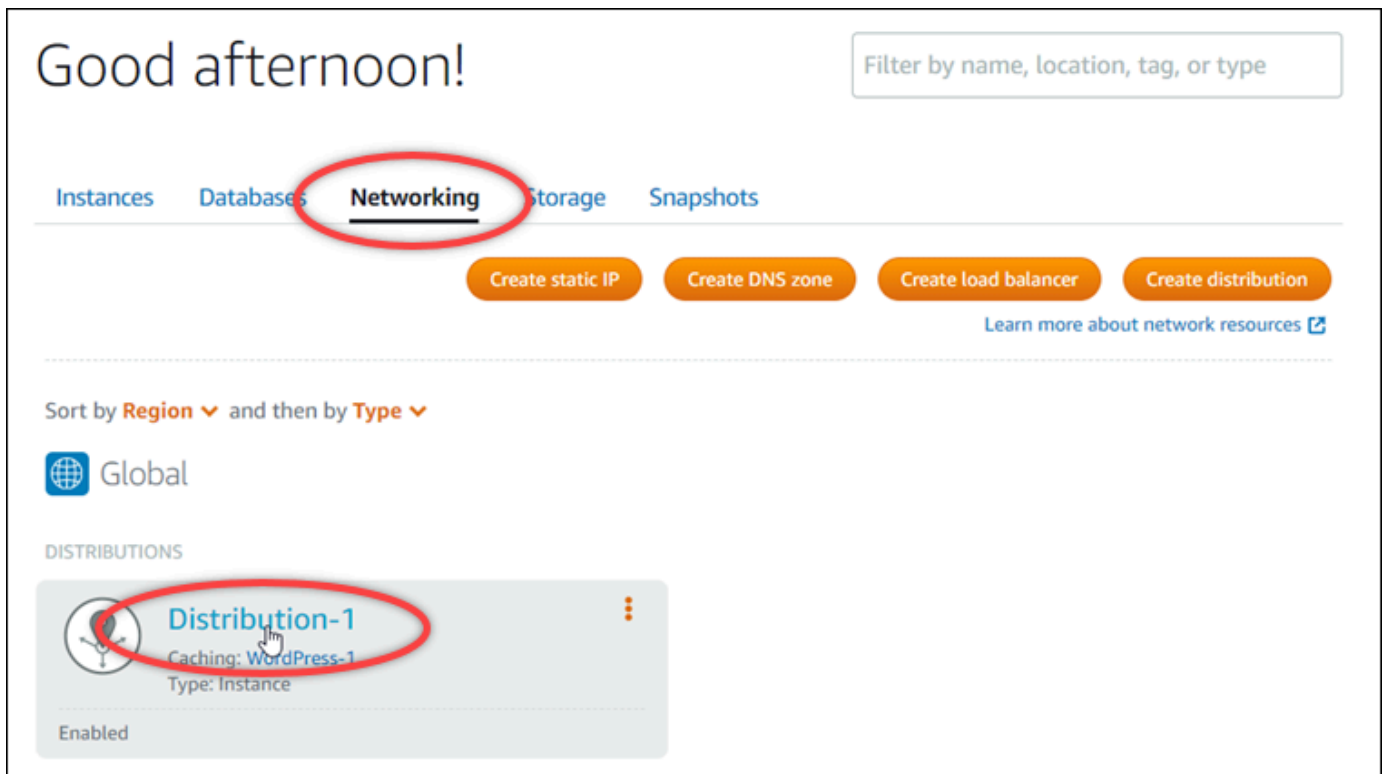
Prerequisite

Before you get started, you need to create an SSL/TLS certificate for your distribution. For more information, see [Create SSL/TLS certificates for your distribution](#).

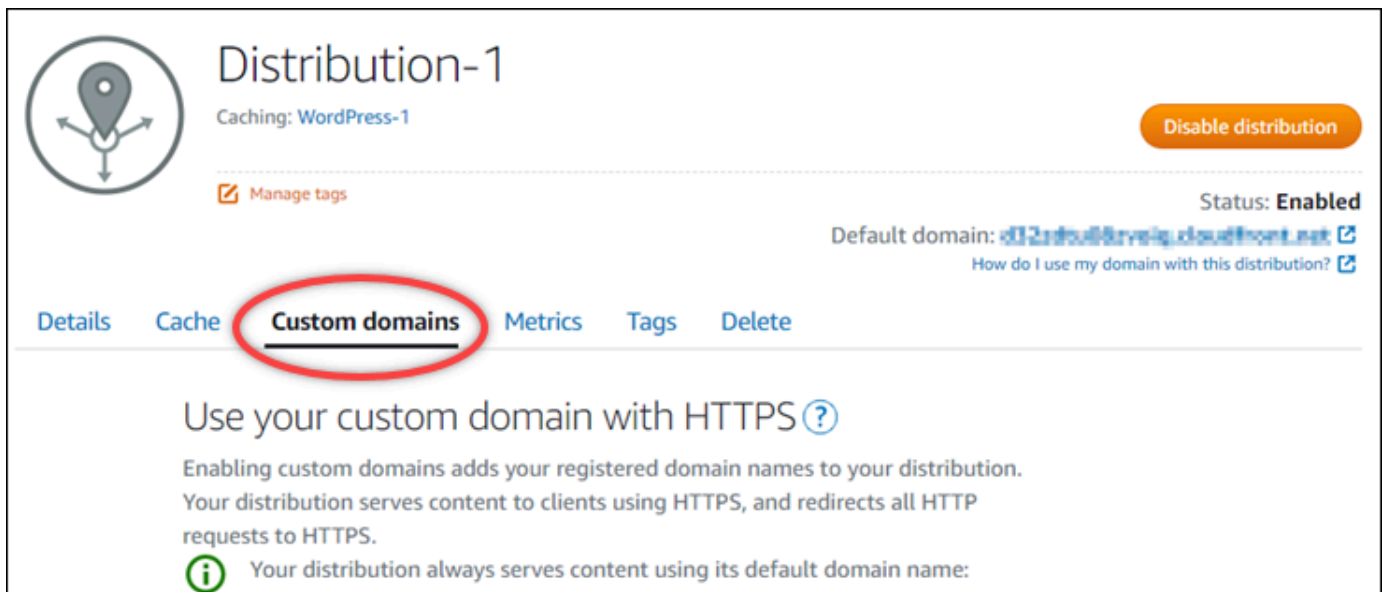
Get the CNAME record values to validate your certificate

Complete the following procedure to get the CNAME records that you must add to your domains to validate the certificate.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which want to get the CNAME record values of a certificate.



4. Choose the **Custom domains** tab on your distribution's management page.



5. Scroll down to the **Attached certificates** section of the page.

All of your distribution certificates are listed under the **Attached certificates** section of the page, including certificates created for other Lightsail resources, and certificates that are pending validation.

6. Find the certificate that you want to validate, expand **Validation details**, and make note of the **Name** and **Value** of the CNAME records that you must add for each domain listed.

You must add these records exactly as listed. We recommend that you copy and paste these values into a text file that you can refer to later. For more information, see the following [Add the CNAME records to your domain's DNS zone](#) section of this guide.

Add the CNAME records to your domain's DNS zone

Complete the following procedure to add CNAME records to your domain's DNS zone.

1. On the Lightsail home page, choose the **Domains & DNS** tab.
2. Under the **DNS zones** section of the page, choose the domain name to which you want to add the CNAME records to validate your certificate.
3. Choose the **DNS records** tab.
4. Choose **Add record** in the DNS records management page.
5. Choose **CNAME** in the **Record type** drop-down.
6. In the **Record name** text box, enter the **Name** value of the CNAME record that you got from your certificate.

The Lightsail console pre-populates the apex portion of your domain. For example, if you want to add the `www.example.com` subdomain, then you only have to enter `www` into the text box, and Lightsail adds the `.example.com` portion for you when you save the record.

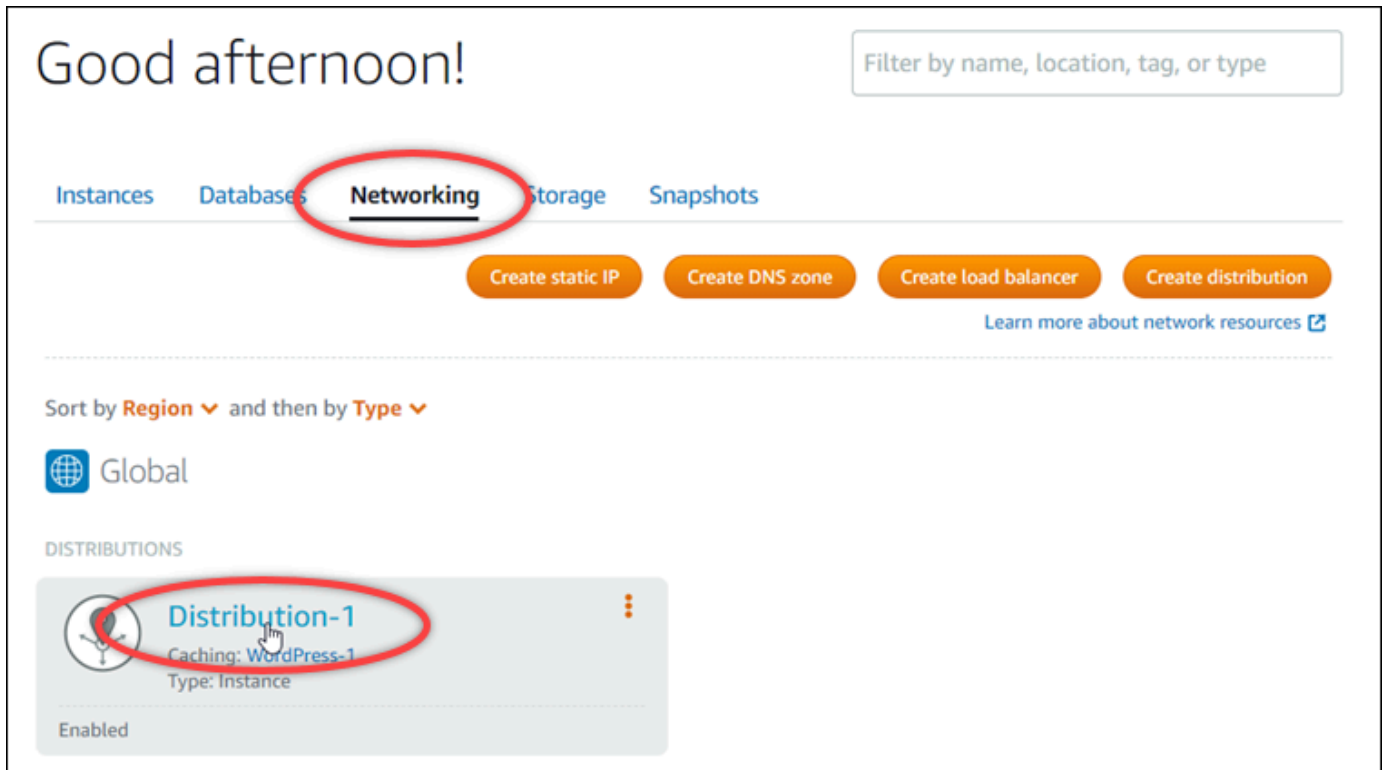
7. In the **Route traffic to** text box, enter the **Value** portion of the CNAME record that you got from your certificate.
8. Confirm that the values you entered are exactly as they were listed on the certificate that you want to validate.
9. Choose the save icon to save the record to your DNS zone.

Repeat these steps to add additional CNAME records for domains on your certificate that need to be validated. Allow time for changes to propagate through the internet's DNS. After a few minutes, you should see if the status of your distribution certificate has changed to **Valid**. For more information, see the following [View the status of your distribution certificate](#) section of this guide.

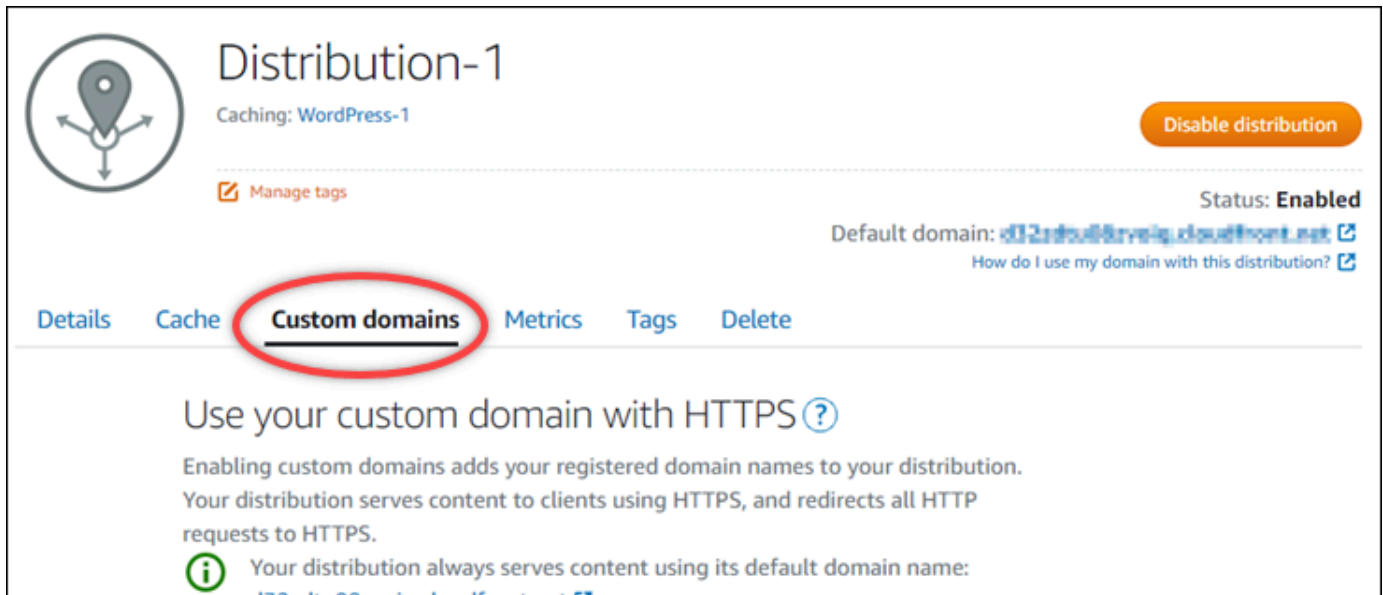
View the status of your distribution certificate

Complete the following procedure to view the status of your SSL/TLS certificate for your distribution.

1. On the Lightsail home page, choose the **Networking** tab.
2. Choose the name of the distribution for which you want to view a certificate's status.

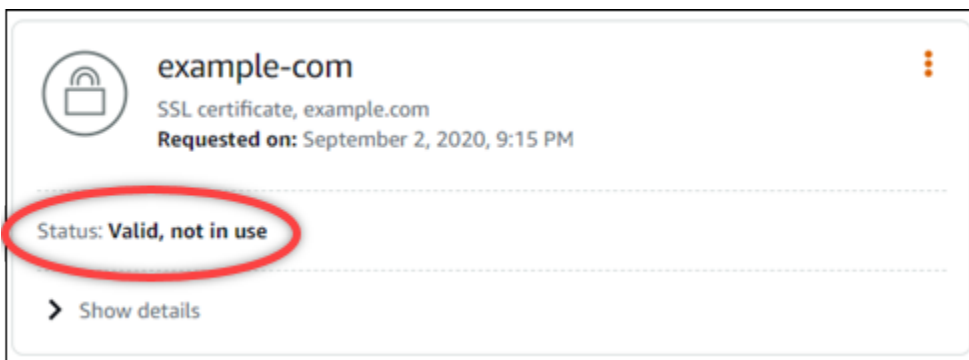


3. Choose the **Custom domains** tab on your distribution's management page.



4. Scroll down to the **Attached certificates** section of the page.

All of your distribution certificates are listed under the **Attached certificates** section of the page, including certificates with **Pending validation** and **Valid** statuses.



A **Valid** status confirms that you successfully validated your certificate with the CNAME records that you added to your domains. Choose **Details** to view your certificate's important dates, encryption details, identification, and validation records. Your certificates are valid for 13 months from the date on which you validated them, after which time Lightsail attempts to automatically re-validate them. Don't delete the CNAME records that you added to your domain because they are required when your certificate is re-validated on the **Valid until** date listed.

After you validate your SSL/TLS certificate, you should enable custom domains for your distribution to use the domain names of your certificate on your distribution. For more information, see [Enable custom domains for your distribution](#).

Configure the minimum TLS protocol version for your Lightsail distribution certificate

Amazon Lightsail uses SSL/TLS certificates to validate custom (registered) domains that you can use with your Lightsail distribution. This guide provides information about the viewer minimum TLS protocol versions (protocol versions) that you can configure for your SSL/TLS certificate. For more information about SSL/TLS certificates, see [SSL/TLS certificates in Lightsail](#). A viewer is an application that makes HTTP requests to the edge locations that are associated to your Lightsail distribution. For more information about distributions, see [Content delivery network distributions in Lightsail](#).

The TLSv1.2_2021 protocol version is configured by default when you enable custom domains for a distribution. You can configure a different protocol version, as described later in this guide. Lightsail distributions do not support custom TLS protocol versions.

Supported protocols

Lightsail distributions can be configured with the following TLS protocols:

- (Recommended) TLSv1.2_2021
- TLSv1.2_2019
- TLSv1.2_2018
- TLSv1.1_2016

Prerequisites

Complete the following prerequisites if you haven't already:

- [Create a Lightsail content delivery network distribution](#)
- [Create SSL/TLS certificates for your distribution](#)
- [Validate SSL/TLS certificates for your distribution](#)
- [Enable custom domains for your distribution](#)
- [Point your domain to the distribution](#)

Identify the minimum TLS protocol version for your distribution

Complete the following steps to identify the minimum TLS protocol version for your Lightsail distribution

Note

In this guide, you will use AWS CloudShell to perform the upgrade. CloudShell is a browser-based, pre-authenticated shell that you can launch directly from the Lightsail console. With CloudShell, you can run AWS CLI commands using your preferred shell, such as Bash, PowerShell, or Z shell. You can do this without downloading or installing command line tools. For more information about how to set up and use CloudShell, see [For more information, see AWS CloudShell in Lightsail](#).

1. Open a Terminal, [AWS CloudShell](#), or Command Prompt window.
2. Enter the following command to identify the minimum TLS protocol version for your Lightsail distribution.

```
aws lightsail get-distributions --distribution-name DistributionName --region us-east-1 | grep "viewerMinimumTlsProtocolVersion"
```

In the command, replace *DistributionName* with the name of the distribution you want to modify.

Example

```
aws lightsail get-distributions --distribution-name Distribution-1 --region us-east-1 | grep "viewerMinimumTlsProtocolVersion"
```

The command will return the ID of the minimum TLS protocol version for your distribution.

Example

```
"viewerMinimumTlsProtocolVersion": "TLSv1.2_2021"
```

Configure the minimum TLS protocol version using the AWS CLI

Complete the following procedure to configure the TLS protocol version using the AWS Command Line Interface (AWS CLI). You do this by using the `update-distribution` command. For more information, see the [update-distribution attribute](#) in the *AWS CLI Command Reference*.

1. Open a Terminal, [AWS CloudShell](#), or Command Prompt window.

2. Enter the following command to change the minimum TLS protocol version for your distribution.

```
aws lightsail update-distribution --distribution-name DistributionName --viewer-  
minimum-tls-protocol-version ProtocolVersion
```

In the command, replace the following example text with your own:

- *DistributionName* with the name of the distribution that you want to update.
- *ProtocolVersion* with the valid TLS protocol version. For example TLSv1.2_2021 or TLSv1.2_2019.

Example:

```
aws lightsail update-distribution --distribution-name MyDistribution --viewer-  
minimum-tls-protocol-version TLSv1.2_2021
```

Your change takes a few moments to become effective.

Delete SSL/TLS certificates for your Lightsail distribution

You can delete Amazon Lightsail SSL/TLS certificates that you're no longer using on your distributions. For example, your certificate might be expired and you've already attached an updated certificate that's validated. For more information about certificates, see [SSL/TLS certificates](#). For more information about distributions, see [Content delivery network distributions](#).

Deleting an SSL/TLS certificate is final and can't be undone. You have a quota of certificates you can create over a 365-day period. For more information, see [Lightsail service quotas](#) in the *AWS General Reference*.

Delete an SSL/TLS certificate for your distribution

Complete the following procedure to delete an SSL/TLS certificate for your distribution.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.

3. Choose the name of the distribution from which you want to delete the SSL/TLS certificate. If the certificate is not currently in use, then you can choose any distribution because all of your certificates are listed in every distribution.
4. Choose the **Custom domains** tab on your distribution's management page.
5. In the **Certificates** section of the page, choose the ellipsis icon (⋮) for the certificate that you want to delete, and choose **Delete**.

The **Delete** option is unavailable if the certificate you want to delete is in use. To delete certificates that are in use, you need to first change the custom domains of the distribution that is using the certificate, or disable custom domains on the distribution that is using the certificate. For more information, see [Change custom domains for your distribution](#) and [Enable custom domains for your distribution](#).

6. Choose **Yes, delete** to confirm the deletion.

Object storage in Amazon Lightsail

Use the Amazon Lightsail object storage service to store and retrieve objects, at any time, from anywhere on the internet. It is designed to make web-scale computing easier for developers, and is built using the Amazon Simple Storage Service (Amazon S3). Lightsail object storage gives you access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of websites. The service aims to maximize benefits of scale and to pass those benefits on to you.

Object storage concepts

The following concepts and terminology apply to Lightsail object storage.

Buckets

A bucket is a container for objects stored in the Lightsail object storage service. Every object is contained in a bucket, which has its own URL. For example, if the object named `media/sailbot.jpg` is stored in the `DOC-EXAMPLE-BUCKET` bucket in the US East (N. Virginia) Region (`us-east-1`), then it is addressable using a URL that is similar to `https://DOC-EXAMPLE-BUCKET.us-east-1.amazonaws.com/media/sailbot.jpg`.

You can create buckets in AWS Regions where Lightsail is available. For more information about which AWS Regions Lightsail is available in, see [Regions and Endpoints](#) in the *AWS General Reference*.

Bucket storage plans

A storage plan, referred to as a *bundle* in the AWS API, specifies the monthly cost, storage space, and data transfer quota for your bucket. You must choose a storage plan when you first create your bucket. You can change it later after your bucket is up and running.

You can change your bucket's plan only one time within your monthly AWS billing cycle. Change your bucket's plan if it's consistently going over its storage space or data transfer quota, or if your bucket's usage is consistently in the lower range of its storage space or data transfer quota. Because your bucket might experience unpredictable usage fluctuations, we strongly recommend that you change your bucket's plan only as a long-term strategy, instead of as a short-term, monthly cost-cutting measure. Choose a storage plan that will provide your bucket with ample storage space and data transfer quotas for a long time to come.

Objects

Objects are the fundamental entities stored in buckets. A file that you upload to your bucket is referred to as an object while it is being stored. Objects consist of *data* and *metadata*. The *data* portion is opaque to the Lightsail object storage service. The *metadata* is a set of name-value pairs that describe the object. These include some default metadata (such as the last modified date), and standard HTTP metadata (such as Content-Type).

An object is uniquely identified within a bucket by a key name and a version ID.

Object key names

A key name is the unique identifier for an object in a bucket. Every object in a bucket has exactly one key. The combination of a bucket, key, and version ID uniquely identifies each object. So you can think of Lightsail object storage as a basic data map between "bucket + key + version" and the object itself. Every object in Lightsail object storage can be uniquely addressed through the combination of the web service endpoint, bucket name, key, and optionally, a version. For example, in the URL `https://DOC-EXAMPLE-BUCKET.us-east-1.amazonaws.com/media/sailbot.jpg`, DOC-EXAMPLE-BUCKET is the name of the bucket and `media/sailbot.jpg` is the object key name.

Object versioning

Versioning is a feature that allows you to keep multiple variants of an object in the same bucket. Enable versioning to preserve, retrieve, and restore every version of every object stored in your bucket. With versioning, you can recover more easily from both unintended user actions and application failures.

Versioning is disabled by default when you create a bucket. After you enable versioning, every version of every object that you store in your bucket is retained until you manually delete the stored version. For example, if you store the `media/sailbot.jpg` object, and later you store a larger file with the same object key name, then the original smaller object is retained as a *previous version*. The new, larger object becomes the *current version*. If you decide that you don't need the previous version of the object, you can delete it. All stored previous versions of an object are deleted when you delete the current version of the object.

Stored object versions consume your bucket's storage space in the same way as stored current versions of an object. After you enable versioning, you can suspend it to stop storing object versions. This also consumes less of your bucket's storage space when you upload new object

versions. When you suspend versioning, stored object versions are retained, but new object versions that you upload while versioning is suspended are not retained.

Bucket and object access

By default, all object storage resources—buckets and objects—are private. This means only the bucket owner, the Lightsail account that created it, can access the bucket and its objects. The bucket owner can optionally grant access permissions to others. This can be done by setting all objects or individual objects to public, which makes them readable to anyone in the world. You can also grant full programmatic access by attaching Lightsail instances to your bucket, or by creating access keys for your bucket. Finally, you can grant other AWS accounts programmatic read-only access to your bucket.

AWS Regions

You can create Lightsail object storage buckets in all of the AWS Regions in which Lightsail is available. You might choose a Region to optimize latency, minimize costs, or address regulatory requirements. Objects stored in an AWS Region do not leave the Region unless you explicitly transfer them to another Region. For example, objects stored in the US West (Oregon) Region do not leave it.

Manage buckets and objects

Lightsail object storage is intentionally built with a minimal feature set that focuses on simplicity and robustness. Following are some of the elements of managing buckets and objects:

- **Create buckets** – Create a bucket that stores data. Buckets are the fundamental containers in the Lightsail object storage service. For more information, see [Create a bucket](#).
- **Store data** – Upload files to your bucket using the Lightsail console, AWS Command Line Interface (AWS CLI), and AWS APIs. For more information about uploading files, see [Upload files to a bucket](#).
- **Download data** – Download your stored objects anytime you want. For more information, see [Download objects from a bucket](#).
- **Grant access** – Grant or deny access to others (such as software or individuals), who want to upload data or download data that is in your bucket. Authentication mechanisms can help keep data secure from unauthorized access. For more information, see [Bucket permissions](#).
- **Manage versioning** – Enable versioning to retain every version of every object stored in your bucket. For more information, see [Enable and suspend object versioning in a bucket](#).

- **Monitor usage** – Monitor the number of objects stored in your bucket, and the amount of storage space being used. For more information, see [View bucket metrics](#).
- **Change the storage plan** – Upsize your bucket if it's being over-utilized, or downsize it if it's being under-utilized. For more information, see [Change the plan of your bucket](#).
- **Connect your bucket** – Connect your Lightsail bucket to your WordPress website to store website images and attachments. You can also specify your bucket as the origin of a Lightsail content delivery network (CDN) distribution. This speeds up the delivery of objects in your bucket to your users around the world. For more information, see [Tutorial: Connect a bucket to your WordPress instance](#) and [Tutorial: Use a bucket with a content delivery network distribution](#).
- **Delete your bucket** – Delete your bucket if you are no longer using it. For more information, see [Delete a bucket](#).

Create a Lightsail bucket

Create a bucket in the Amazon Lightsail object storage service when you're ready to start uploading your files to the cloud. Every file that you upload to the Lightsail object storage service is stored in a Lightsail bucket. For more information about buckets, see [Object storage](#).

Create a bucket

Complete the following procedure to create a Lightsail bucket.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose **Create bucket**.
4. Choose **Change AWS Region** to choose the Region in which to create your bucket.

We recommend that you create your bucket in the same AWS Region as the resources that you plan to use with your bucket. You cannot change the Region of your bucket after you create it.

5. Choose a storage plan for your bucket.

The storage plan specifies the monthly cost, storage space quota, and data transfer quota for your bucket.

You can change your bucket's plan only one time within your monthly AWS billing cycle.

Change your bucket's plan if it's consistently going over its storage space or data transfer

quota, or if your bucket's usage is consistently in the lower range of its storage space or data transfer quota. For more information see [Change the plan of your bucket](#).

6. Enter a name for your bucket.

For more information about bucket names, see [Bucket naming rules in Amazon Lightsail](#).

7. Choose **Create bucket**.

You are redirected to the management page of your new bucket. Continue to the Next steps section of this guide for additional documentation to use and manage your bucket.

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
- [Configuring bucket access permissions in Amazon Lightsail](#)
- [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
- [Creating access keys for a bucket in Amazon Lightsail](#)
- [Configuring resource access for a bucket in Amazon Lightsail](#)
- [Configuring cross-account access for a bucket in Amazon Lightsail](#)

5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).

14 Learn how to connect your bucket to other resources. For more information, see the following tutorials.

- [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
- [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)

15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Delete a Lightsail bucket

Delete your bucket in the Amazon Lightsail object storage service if you're no longer using it. When you delete your bucket, all objects in the bucket, including stored versions of objects and access keys, are permanently deleted.

For more information about buckets, see [Object storage](#).

Force deleting a bucket

Buckets that have one of the following conditions cannot be deleted unless you acknowledge the deletion:

- The bucket is the origin of a distribution.
- The bucket has instances attached to it.
- The bucket has objects.
- The bucket has access keys.

You must acknowledge the deletion to ensure that you don't disrupt an existing workflow that relies on the bucket. For example, a WordPress website that is storing media on the bucket or a distribution that is caching and serving objects in your bucket.

To acknowledge deletion of a bucket that has one of the preceding conditions, you must force delete the bucket. Before you delete the bucket, the Lightsail service prompts you about which of these conditions exist on it. If you use the Lightsail console to delete your bucket, you are presented with the option to force delete it. If you use the AWS CLI, you must specify the `--force-delete` flag when making a `delete-bucket` request. Both of these procedures are

covered in the [Delete your bucket using the Lightsail console](#) and [Delete your bucket using the AWS CLI](#) sections of this guide.

Delete your bucket using the Lightsail console

Complete the following procedure to delete your bucket using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket that you want to delete.
4. Choose the ellipsis (:) icon in the tab menu, then choose **Delete**.
5. Choose **Delete bucket**.
6. In the prompt that appears, confirm if your bucket meets any of the following conditions:
 - Contains an object
 - Has access keys
 - Is attached to an instance
 - Is the origin of a distribution

If it has any of those conditions, then you must choose to force delete the bucket.

7. Choose one of the following options:
 - Choose **Force delete** to delete your bucket even if it has any of the conditions listed in step 6 of this procedure.
 - Choose **Yes, delete** to delete your bucket when it doesn't have any of the conditions listed in step 6 of this procedure.
 - Choose **No, cancel** to cancel deletion.

Delete your bucket using the AWS CLI

Complete the following procedure to delete your bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `delete-bucket` command. For more information, see [delete-bucket](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. In the command prompt or terminal window, enter one of the following commands:
 - Enter the following command to delete a bucket that doesn't have the conditions listed in the [Force deleting a bucket](#) section of this guide.

```
aws lightsail delete-bucket --bucket-name BucketName
```

- Enter the following command to force delete a bucket that has the conditions listed in the [Force deleting a bucket](#) section of this guide.

```
aws lightsail delete-bucket --bucket-name BucketName --force-delete
```

In the commands, replace *BucketName* with the name of the bucket you want to delete.

Example:

```
aws lightsail delete-bucket --bucket-name DOC-EXAMPLE-BUCKET
```

You should see a result similar to the following example:


```
C:\>aws lightsail delete-bucket --bucket-name DOC-EXAMPLE-BUCKET
{
  "operations": [
    {
      "id": "6example-4d30-4442-ae9a-examplef4f52",
      "resourceName": "DOC-EXAMPLE-BUCKET",
      "resourceType": "Bucket",
      "createdAt": "2021-06-30T13:42:43.873000-07:00",
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "62example362/DOC-EXAMPLE-BUCKET/small_1_0",
      "operationType": "DeleteBucket",
      "status": "Succeeded",
      "statusChangedAt": "2021-06-30T13:42:43.873000-07:00",
      "errorCode": "",
      "errorDetails": ""
    }
  ]
}
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
- [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
- [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in](#)

- 10 After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
- 11 Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
- 12 Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
- 13 Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
- 14 Learn how to connect your bucket to other resources. For more information, see the following tutorials.
 - [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
- 15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Create Lightsail bucket access keys

Use access keys to create a set of credentials that grant full access to a bucket and its objects. You can configure access keys on your software or plugin so that it can have full read and write access to a bucket using the AWS APIs, and AWS SDKs. You can also configure access keys on the AWS CLI.

Access keys consist of an access key ID and a secret access key as a set. The secret access key is visible only when you create it. If your secret access key is copied, is lost, or becomes compromised, you should delete your access key and create a new one. You can have a maximum of two access keys per bucket. Even though you can have two, having one access key for your bucket is useful when you need to rotate the key. To rotate an access key, create a new one, configure it on your software and test it, and then delete the earlier key. After you delete an access key, it's gone forever and can't be restored. It can only be replaced with a new access key.

For more information about permission options, see [Bucket permissions](#). For more information about security best practices, see [Security Best Practices for object storage](#). For more information about buckets, see [Object storage](#).

Create access keys for a bucket

Complete the following procedure to create access keys for a bucket.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to configure access permissions.
4. Choose the **Permissions** tab.

The **Access keys** section of the page displays the existing access keys for the bucket, if any.

5. Choose **Create access key** to create a new access key for the bucket.

Note

You can also choose to delete an existing access key by choosing the trash bin icon for the key you want to delete.

6. In the prompt that appears, choose **Yes, create** to confirm that you want to create a new access key. Otherwise, choose **No, cancel**.
7. In the success prompt that appears, make a note of the access key ID.
8. Choose **Show secret access key** to view the secret access key, and make a note of it. The secret access key will not be shown again.

Important

Store your access key ID and secret access key in a secure location. If it becomes compromised, you should delete it and create a new one.

9. Choose **Continue** to finish.

The new access key is listed in the **Access keys** section of the page. If your access key becomes compromised, or lost, delete it and create a new one.

Note

The **Last used** column displayed next to each access key identifies when the key was last used. A dash is displayed when the key has not been used. Expand the access key node to view the service and AWS Region where the key was last used.

Block public access for Lightsail buckets

Amazon Simple Storage Service (Amazon S3) is an object storage service on which customers can store and protect data. The Amazon Lightsail object storage service is built on Amazon S3 technology. Amazon S3 offers *account-level block public access*, which you can use to limit public access to all S3 buckets in an AWS account. Account-level block public access can make all S3 buckets in an AWS account private, regardless of existing individual bucket and object permissions.

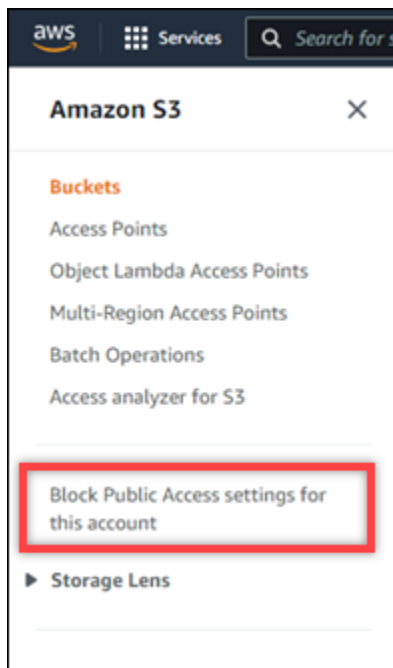
When allowing or denying public access, Lightsail object storage buckets take into account the following:

- Lightsail bucket access permissions. For more information see [Bucket permissions](#).
- Amazon S3 account-level block public access configurations, which override the Lightsail bucket access permissions.

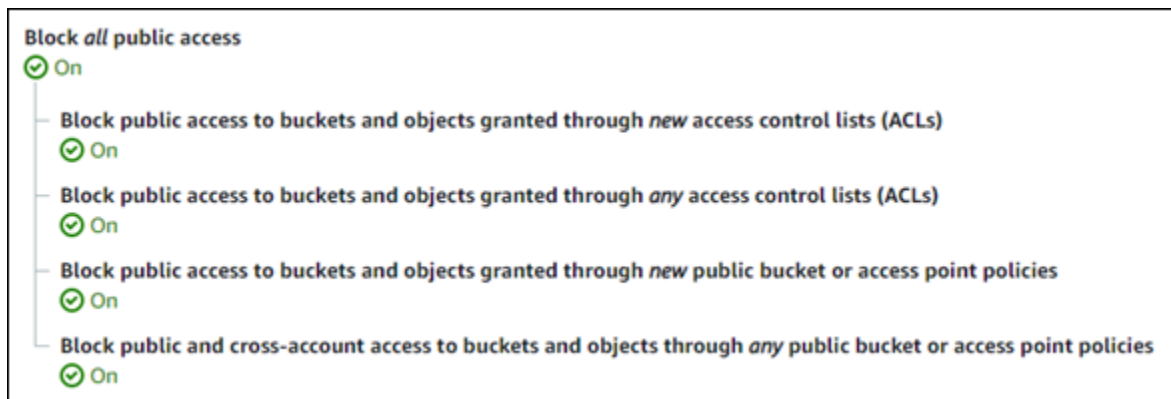
If you turn on account-level **Block all public access** in Amazon S3, your public Lightsail buckets and objects become private and are no longer publicly accessible.

Configuring block public access settings for your account

You can use the Amazon S3 console, AWS Command Line Interface (AWS CLI), AWS SDKs, and REST API to configure block public access settings. You can access the account-level block public access feature in the navigation pane of the Amazon S3 console as shown in the following example.



The Amazon S3 console offers settings to block all public access, block public access granted through new or any access control lists, and block public access to buckets and objects granted through new or any public bucket or access point policies.




You can turn each setting **On** or **Off** in the Amazon S3 console. In the API, the corresponding setting is TRUE (On) or FALSE (Off). The following sections describe each setting's effects on S3 buckets and Lightsail buckets.

Note

The following sections mention access control lists (ACLs). An ACL defines the users who own or have access to a bucket or individual objects. For more information, see [Access control list overview](#) in the *Amazon S3 User Guide*.

- **Block *all* public access** — Turn on this setting to block all public access to your S3 buckets, Lightsail buckets, and their corresponding objects. This setting incorporates all of the following settings. When you turn on this setting, only you (the bucket owner) and authorized users are allowed to access your buckets and their objects. You can only turn this setting on in the Amazon S3 console. It is not available in the AWS CLI, Amazon S3 API, or AWS SDKs.
- **Block public access to buckets and objects granted through *new* access control lists (ACLs)** — Turn on this setting to block putting public ACLs on buckets and objects. This setting does not impact existing ACLs. Therefore, an object that already has a public ACL remains public. This setting also has no impact on objects that are public due to a bucket access permission being set to **All objects are public and read-only**. This setting is labeled as `BlockPublicAcls` in the Amazon S3 API.

 **Note**

WordPress plugins that put media in Lightsail buckets, such as the Offload Media Light plugin, might stop working when this setting is turned on. This is because most WordPress plugins configure the public-read ACL on objects. WordPress plugins that toggle object ACLs might also stop working.

- **Block public access to buckets and objects granted through *any* access control lists (ACLs)** — Turn on this setting to ignore public ACLs and block public access to buckets and objects. This setting allows public ACLs to be put on buckets and objects, but ignores them when granting access. For Lightsail buckets, setting a bucket's access permission to **All objects are public and read-only** or setting an individual object's permission to **Public (read-only)** is the equivalent of putting a public ACL on either. This setting is labeled as `IgnorePublicAcls` in the Amazon S3 API.
- **Block public access to buckets and objects granted through *new* public bucket or access point policies** — Turn on this setting to block the **All objects are public and read-only** bucket access permission from being configured on your Lightsail buckets. This setting does not impact buckets that are already configured with the **All objects are public and read-only** bucket access permission. This setting is labeled as `BlockPublicPolicy` in the Amazon S3 API.
- **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies** — Turn on this setting to make all of your Lightsail buckets private. This makes all Lightsail buckets private, even if they are configured with the **All objects are public**

and read-only bucket access permission. This setting is labeled as `RestrictPublicBuckets` in the Amazon S3 API.

Important

This setting also blocks cross-account access that is configured on a Lightsail bucket that is also configured with the **All objects are public and read-only** bucket access permission in Lightsail. To continue allowing cross-account access, make sure to configure the Lightsail bucket with the **All objects are private** bucket access permission in Lightsail before turning on the **Block public and cross-account access to buckets and objects through any public bucket or access point policies** setting in Amazon S3.

For more information about block public access and how to configure it, see the following resources in the *Amazon S3 User Guide*:

- [Blocking public access to your Amazon S3 storage](#)
- [Configuring block public access settings for your account](#)

Use the Lightsail console, AWS CLI, AWS SDKs, and REST API to configure access permissions for your Lightsail buckets. For more information, see [Bucket permissions](#).

Note

Lightsail uses a service-linked role to get the current account-level block public access configuration from Amazon S3 and apply it to Lightsail object storage resources. After configuring block public access in Amazon S3, wait at least one hour for it to take effect in Lightsail. For more information, see [Service-linked roles](#).

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).

2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
 6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
 7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
 8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)

- [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
14. Learn how to connect your bucket to other resources. For more information, see the following tutorials.
- [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
15. Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Bucket access logs in Amazon Lightsail

Access logging provides detailed records for the requests that are made to a bucket in the Amazon Lightsail object storage service. This information can include the request type, the resources that are specified in the request, and the time and date that the request was processed. Access logs are useful for many applications. For example, access log information can be useful in security and access audits. It can also help you learn about your customer base.

Contents

- [What do I need to enable log delivery](#)
- [Log object key format](#)
- [How are logs delivered?](#)
- [Best effort access log delivery](#)
- [Bucket logging status changes take effect over time](#)

What do I need to enable log delivery?

Consider the following before enabling log delivery. For details, see [Enable bucket access logging](#).

1. **Identify the target bucket for the logs.** This bucket is where you want Lightsail to save the access logs as objects. Both the source and target buckets must be in the same AWS Region and owned by the same account.

You can have logs delivered to any bucket that you own that is in the same Region as the source bucket, including the source bucket itself. But for simpler log management, we recommend that you save access logs in a different bucket.

When your source bucket and target bucket are the same bucket, additional logs are created for the logs that are written to the bucket. This might not be ideal because it could result in a small increase in your storage consumption. In addition, the extra logs about logs might make it harder to find the log that you are looking for. If you choose to save access logs in the source bucket, we recommend that you specify a prefix for the log object keys so that the object names begin with a common string and the log objects are easier to identify. Key prefixes are also useful to distinguish between source buckets when multiple buckets log to the same target bucket.

2. **(Optional) Identify a prefix for the log object keys.** The prefix makes it simpler for you to locate the log objects. For example, if you specify the prefix value `logs/`, each log object that Lightsail creates begins with the `logs/` prefix in its key. The trailing slash `/` is required to denote the end of the prefix. Following is an example of a log object key with the `logs/` prefix:

```
logs/2021-11-31-21-32-16-E568B2907131C0C0
```

Log object key format

Lightsail uses the following object key format for the log objects it uploads in the target bucket:

```
TargetPrefix/YYYY-mm-DD-HH-MM-SS-UniqueString
```

In the key, YYYY, mm, DD, HH, MM, and SS are the digits of the year, month, day, hour, minute, and seconds (respectively) when the log file was delivered. These dates and times are in Coordinated Universal Time (UTC).

A log file delivered at a specific time can contain records written at any point before that time. There is no way to know whether all log records for a certain time interval have been delivered or not.

The UniqueString component of the key is there to prevent overwriting of files. It has no meaning, and log processing software should ignore it.

How are logs delivered?

Lightsail periodically collects access log records, consolidates the records in log files, and then uploads log files to your target bucket as log objects. If you enable logging on multiple source buckets that deliver to the same target bucket, the target bucket will have access logs for all those source buckets. However, each log object reports access log records for a specific source bucket.

Best effort access log delivery

Access log records are delivered on a best effort basis. Most requests for a bucket that is properly configured for logging result in a delivered log record. Most log records are delivered within a few hours of the time that they are recorded, but they can be delivered more frequently.

The completeness and timeliness of access logging is not guaranteed. The log record for a particular request might be delivered long after the request was actually processed, or it might not be delivered at all. The purpose of access logs is to give you an idea of the nature of traffic against your bucket. It is rare to lose log records, but access logging is not meant to be a complete accounting of all requests.

Bucket logging status changes take effect over time

Changes to the logging status of a bucket take time to actually affect the delivery of log files. For example, if you enable logging for a bucket, some requests made in the following hour might be logged, while others might not. If you change the target bucket for logging from bucket A to bucket B, some logs for the next hour might continue to be delivered to bucket A, while others

might be delivered to the new target bucket B. In all cases, the new settings eventually take effect without any further action on your part.

Topics

- [Bucket access log formatting in Amazon Lightsail](#)
- [Enable bucket access logging in Amazon Lightsail](#)
- [Use bucket access logs to identify requests in Amazon Lightsail](#)

Bucket access log formatting in Amazon Lightsail

Access logging provides detailed records for the requests that are made to a bucket in the Amazon Lightsail object storage service. You can use access logs for security and access audits, or learn about your customer base. This section describes the format and other details about access log files. For more information about logging basics, see [Bucket access logs](#).

Access log files consist of a sequence of newline-delimited log records. Each log record represents one request and consists of space-delimited fields.

The following is an example log consisting of five log records.

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
REST.GET.VERSIONING - "GET /awsexamplebucket1?versioning HTTP/1.1" 200 - 113 - 7 -
 "-" "S3Console/0.4" - s9lzhYrFp76ZVxRcpX9+5cjAnEH2R0uNkd2BHfIa6UkFVdtjF5mKR3/eTPFvsiP/
XV/VLi31234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader awsexamplebucket1.s3.us-
west-1.amazonaws.com TLSV1.1
```

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
REST.GET.LOGGING_STATUS - "GET /awsexamplebucket1?logging HTTP/1.1" 200 - 242
- 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mX0cqPGzQ0I5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBU0ZJG96N1234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader
awsexamplebucket1.s3.us-west-1.amazonaws.com TLSV1.1
```

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
```

```
REST.GET.BUCKETPOLICY - "GET /awsexamplebucket1?policy HTTP/1.1" 404
NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNaBsXZQQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeMC78iwEIWxs99CRUrbS4n11234= SigV2 ECDHE-RSA-AES128-GCM-SHA256
AuthHeader awsexamplebucket1.s3.us-west-1.amazonaws.com TLSV1.1
```

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:01:00 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
REST.GET.VERSIONING - "GET /awsexamplebucket1?versioning HTTP/1.1" 200 - 113
- 33 - "-" "S3Console/0.4" - Ke1bUcazaN1jWuU1PJaxF64cQVpUEhoZKEG/hmy/gijN/
I1DeWqDfFvnpYbfEseEME/u7ME1234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader
awsexamplebucket1.s3.us-west-1.amazonaws.com TLSV1.1
```

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:01:57 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /awsexamplebucket1/
s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-" "S3Console/0.4" -
10S62Zv81kBW7BB6SX4XJ48o6kpc16LPwEoizZQxJd5qDSCTLX0TgS37kYUBKQW3+bPdrg1234= SigV4
ECDHE-RSA-AES128-SHA AuthHeader awsexamplebucket1.s3.us-west-1.amazonaws.com TLSV1.1
```

Note

Any log record field can be set to – (dash) to indicate that the data was unknown or unavailable, or that the field was not applicable to the request.

Contents

- [Log record fields](#)
- [Additional logging for copy operations](#)
- [Custom access log information](#)
- [Programming considerations for extensible access log format](#)

Log record fields

The following list describes the log record fields.

Access Point ARN (Amazon Resource Name)

The Amazon Resource Name (ARN) of the access point of the request. If access point ARN is malformed or not used, the field will contain a '-'. For more information on access points, see [Using access points](#). For more information on ARNs, see the topic on [Amazon Resource Name \(ARN\)](#) in the *AWS General Reference*.

Example entry

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

Bucket Owner

The canonical user ID of the owner of the source bucket. The canonical user ID is another form of the AWS account ID. For more information about the canonical user ID, see [AWS account identifiers](#) in the *AWS General Reference*. For information about how to find the canonical user ID for your account, see [Finding the canonical user ID for your AWS account](#).

Example entry

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

The name of the bucket that the request was processed against. If the system receives a malformed request and cannot determine the bucket, the request will not appear in any access log.

Example entry

```
awsexamplebucket1
```

Time

The time at which the request was received; these dates and times are in Coordinated Universal Time (UTC). The format, using *strftime()* terminology, is as follows: `[%d/%b/%Y:%H:%M:%S %z]`

Example entry

```
[06/Feb/2019:00:00:38 +0000]
```

Remote IP

The apparent internet address of the requester. Intermediate proxies and firewalls might obscure the actual address of the machine making the request.

Example entry

```
192.0.2.3
```

Requester

The canonical user ID of the requester, or a - for unauthenticated requests. If the requester was an IAM user, this field returns the requester's IAM user name along with the AWS root account that the IAM user belongs to. This identifier is the same one used for access control purposes.

Example entry

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Request ID

A string generated by Lightsail to uniquely identify each request.

Example entry

```
3E57427F33A59F07
```

Operation

The operation listed here is declared as SOAP . *operation*, REST . *HTTP_method.resource_type*, WEBSITE . *HTTP_method.resource_type*, or BATCH . DELETE . OBJECT.

Example entry

```
REST.PUT.OBJECT
```

Key

The "key" part of the request, URL encoded, or "-" if the operation does not take a key parameter.

Example entry


```
/photos/2019/08/puppy.jpg
```

Request-URI

The Request-URI part of the HTTP request message.

Example Entry

```
"GET /awsexamplebucket1/photos/2019/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

HTTP status

The numeric HTTP status code of the response.

Example entry

```
200
```

Error Code

The Amazon S3 [Error code](#), or "-" if no error occurred.

Example entry

```
NoSuchBucket
```

Bytes Sent

The number of response bytes sent, excluding HTTP protocol overhead, or "-" if zero.

Example entry

```
2662992
```

Object Size

The total size of the object in question.

Example entry

```
3462992
```

Total Time

The number of milliseconds the request was in flight from the bucket's perspective. This value is measured from the time your request is received to the time that the last byte of the response is sent. Measurements made from the client's perspective might be longer due to network latency.

Example entry

```
70
```

Turn-Around Time

The number of milliseconds that Lightsail spent processing your request. This value is measured from the time the last byte of your request was received until the time the first byte of the response was sent.

Example entry

```
10
```

Referer

The value of the HTTP Referer header, if present. HTTP user-agents (for example, browsers) typically set this header to the URL of the linking or embedding page when making a request.

Example entry

```
"http://www.amazon.com/webservices"
```

User-Agent

The value of the HTTP User-Agent header.

Example entry

```
"curl/7.15.1"
```

Version Id

The version ID in the request, or - if the operation does not take a `versionId` parameter.

Example entry

```
3HL4kqtJvjVBH40Nıjfkđ
```

Host Id

The `x-amz-id-2` or Lightsail extended request ID.

Example entry

```
s9lzhYıFp76ZVxRcpX9+5cjAnEH2R0uNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Signature Version

The signature version, `SigV2` or `SigV4`, that was used to authenticate the request or a - for unauthenticated requests.

Example entry

```
SigV2
```

Cipher Suite

The Secure Sockets Layer (SSL) cipher that was negotiated for HTTPS request or a - for HTTP.

Example entry

```
ECDHE-RSA-AES128-GCM-SHA256
```

Authentication Type

The type of request authentication used, `AuthHeader` for authentication headers, `QueryString` for query string (pre-signed URL) or a - for unauthenticated requests.

Example entry

```
AuthHeader
```

Host Header

The endpoint used to connect to Lightsail.

Example entry

```
s3.us-west-2.amazonaws.com
```

TLS version

The Transport Layer Security (TLS) version negotiated by the client. The value is one of following: TLSv1, TLSv1.1, TLSv1.2; or - if TLS wasn't used.

Example entry

```
TLSv1.2
```

Additional logging for copy operations

A copy operation involves a GET and a PUT. For that reason, we log two records when performing a copy operation. The previous section describes the fields related to the PUT part of the operation. The following list describes the fields in the record that relate to the GET part of the copy operation.

Bucket Owner

The canonical user ID of the bucket that stores the object being copied. The canonical user ID is another form of the AWS account ID. For more information about the canonical user ID, see [AWS account identifiers](#) in the *AWS General Reference*. For information about how to find the canonical user ID for your account, see [Finding the canonical user ID for your AWS account](#).

Example entry

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

The name of the bucket that stores the object being copied.

Example entry

```
awsexamplebucket1
```

Time

The time at which the request was received; these dates and times are in Coordinated Universal time (UTC). The format, using `strftime()` terminology, is as follows: [%d/%B/%Y:%H:%M:%S %z]

Example entry

```
[06/Feb/2019:00:00:38 +0000]
```

Remote IP

The apparent internet address of the requester. Intermediate proxies and firewalls might obscure the actual address of the machine making the request.

Example entry

```
192.0.2.3
```

Requester

The canonical user ID of the requester, or a - for unauthenticated requests. If the requester was an IAM user, this field will return the requester's IAM user name along with the AWS root account that the IAM user belongs to. This identifier is the same one used for access control purposes.

Example entry

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Request ID

A string generated by Lightsail to uniquely identify each request.

Example entry

```
3E57427F33A59F07
```

Operation

The operation listed here is declared as SOAP . *operation*, REST . *HTTP_method.resource_type*, WEBSITE . *HTTP_method.resource_type*, or BATCH . DELETE . OBJECT .

Example entry

```
REST.COPY.OBJECT_GET
```

Key

The "key" of the object being copied or "-" if the operation does not take a key parameter.

Example entry

```
/photos/2019/08/puppy.jpg
```

Request-URI

The Request-URI part of the HTTP request message.

Example entry

```
"GET /awsexamplebucket1/photos/2019/08/puppy.jpg?x-foo=bar"
```

HTTP status

The numeric HTTP status code of the GET portion of the copy operation.

Example entry

```
200
```

Error Code

The Amazon S3 Error code, of the GET portion of the copy operation or - if no error occurred.

Example entry

```
NoSuchBucket
```

Bytes Sent

The number of response bytes sent, excluding HTTP protocol overhead, or "-" if zero.

Example entry

```
2662992
```

Object Size

The total size of the object in question.

Example entry

```
3462992
```

Total Time

The number of milliseconds the request was in flight from the bucket's perspective. This value is measured from the time your request is received to the time that the last byte of the response is sent. Measurements made from the client's perspective might be longer due to network latency.

Example entry

```
70
```

Turn-Around Time

The number of milliseconds that Lightsail spent processing your request. This value is measured from the time the last byte of your request was received until the time the first byte of the response was sent.

Example entry

```
10
```

Referer

The value of the HTTP Referer header, if present. HTTP user-agents (for example, browsers) typically set this header to the URL of the linking or embedding page when making a request.

Example entry

```
"http://www.amazon.com/webservices"
```

User-Agent

The value of the HTTP User-Agent header.

Example entry

```
"curl/7.15.1"
```

Version Id

The version ID of the object being copied or - if the `x-amz-copy-source` header didn't specify a `versionId` parameter as part of the copy source.

Example entry

```
3HL4kqtJvjVBH40NıjfkD
```

Host Id

The `x-amz-id-2` or Lightsail extended request ID.

Example entry

```
s91zHYıFp76ZVxRcpX9+5cjAnEH2R0uNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Signature Version

The signature version, `SigV2` or `SigV4`, that was used to authenticate the request or a - for unauthenticated requests.

Example entry

```
SigV2
```

Cipher Suite

The Secure Sockets Layer (SSL) cipher that was negotiated for HTTPS request or a - for HTTP.

Example entry

```
ECDHE-RSA-AES128-GCM-SHA256
```

Authentication Type

The type of request authentication used, `AuthHeader` for authentication headers, `QueryString` for query string (presigned URL) or `-` for unauthenticated requests.

Example entry

```
AuthHeader
```

Host Header

The endpoint used to connect to Lightsail.

Example entry

```
s3.us-west-2.amazonaws.com
```

TLS version

The Transport Layer Security (TLS) version negotiated by the client. The value is one of following: `TLSv1`, `TLSv1.1`, `TLSv1.2`; or `-` if TLS wasn't used.

Example entry

```
TLSv1.2
```

Custom access log information

You can include custom information to be stored in the access log record for a request. To do this, add a custom query-string parameter to the URL for the request. Lightsail ignores query-string parameters that begin with "x-", but includes those parameters in the access log record for the request, as part of the `Request-URI` field of the log record.

For example, a GET request for `"s3.amazonaws.com/awsexamplebucket1/photos/2019/08/puppy.jpg?x-user=johndoe"` works the same as the request for `"s3.amazonaws.com/`

awsexamplebucket1/photos/2019/08/puppy.jpg", except that the "x-user=johndoe" string is included in the Request-URI field for the associated log record. This functionality is available in the REST interface only.

Programming considerations for extensible access log format

Occasionally we might extend the access log record format by adding new fields to the end of each line. Therefore, you should write any code that parses access logs to handle trailing fields that it might not understand.

Enable bucket access logging in Amazon Lightsail

Access logging provides detailed records for the requests that are made to a bucket in the Amazon Lightsail object storage service. Access logs are useful for many applications. For example, access log information can be useful in security and access audits. It can also help you learn about your customer base.

By default, Lightsail doesn't collect access logs for your buckets. When you enable logging, Lightsail delivers access logs for a source bucket to a target bucket that you choose. Both the source and target buckets must be in the same AWS Region and owned by the same account.

An access log record contains details about the requests that are made to a bucket. This information can include the request type, the resources that are specified in the request, and the time and date that the request was processed. In this guide, we show you how to enable or disable access logging for your buckets by using the Lightsail API, the AWS Command Line Interface (AWS CLI), or AWS SDKs.

For more information about logging basics, see [Bucket access logs](#).

Contents

- [Costs for access logging](#)
- [Enable access logging using the AWS CLI](#)
- [Disable access logging using the AWS CLI](#)

Costs for access logging

There is no extra charge for enabling access logging on a bucket. However, log files that the system delivers to a bucket will use up storage space. You can delete the log files at any time. We do not

assess data transfer charges for log file delivery when the log bucket's data transfer is within its configured monthly allowance.

Your target bucket should not have access logging enabled. You can have logs delivered to any bucket that you own that is in the same Region as the source bucket, including the source bucket itself. However, for simpler log management, we recommend that you save access logs in a different bucket.

Enable access logging using the AWS CLI

To enable access logging for your buckets, we recommend that you create a dedicated logging bucket in each AWS Region that you have buckets. Then have the access log delivered to that dedicated logging bucket.

Complete the following procedure to enable access logging using the AWS CLI.

Note

You must install the AWS CLI and configure it for Lightsail before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window on your local computer.
2. Enter the following command to enable access logging.

```
aws lightsail update-bucket --bucket-name SourceBucketName --access-log-config
{"\"enabled\": true, \"destination\": \"TargetBucketName\", \"prefix\":
\"ObjectKeyNamePrefix/\"}"
```

In the command, replace the following example text with your own:

- *SourceBucketName* - The name of the source bucket for which the access logs will be created.
- *TargetBucketName* – The name of the target bucket where the access logs will be saved.
- *ObjectKeyNamePrefix/* - The optional object key name prefix for the access logs. Note that the prefix must end with a forward slash (/).

Example

```
aws lightsail update-bucket --bucket-name MyExampleBucket --access-log-config
"{\"enabled\": true, \"destination\": \"MyExampleLogDestinationBucket\", \"prefix
\": \"logs/MyExampleBucket/\"}"
```

In the example, *MyExampleBucket* is the source bucket for which access logs will be created, *MyExampleLogDestinationBucket* is the destination bucket where the access logs will be saved, and *logs/MyExampleBucket/* is the object key name prefix for the access logs.

You should see a result similar to the following example after running the command. The source bucket is updated, and the access logs should begin generating and being stored on the destination bucket.

```
c:\Models>aws lightsail update-bucket --bucket-name MyExampleBucket
--access-log-config "{\"enabled\": true, \"destination\": \"MyExampleLogDestinationBucket\", \"prefix\": \"logs/MyExampleBucket/\"}"

{
  "bucket": {
    "resourceType": "Bucket",
    "accessRules": {
      "getObject": "private",
      "allowPublicOverrides": false
    },
    "arn": "arn:aws:s3:::MyExampleBucket",
    "bundleId": "large_1_0",
    "createdAt": "2021-06-29T08:12:39.163000-07:00",
    "url": "https://MyExampleBucket.s3.amazonaws.com",
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "name": "MyExampleBucket",
    "supportCode": "MyExampleBucket",
    "tags": [],
    "objectVersioning": "Suspended",
    "ableToUpdateBundle": true,
    "readonlyAccessAccounts": [
      "MyExampleAccount"
    ],
    "state": {
      "code": "OK"
    },
    "accessLogConfig": {
      "enabled": true,
      "destination": "MyExampleLogDestinationBucket"
      "prefix": "logs/MyExampleBucket/"
    }
  },
  "operations": [
    {
      "id": "7ee31ae9-2946-4889-9083-4b0459538162",
      "resourceName": "MyExampleBucket",
      "resourceType": "Bucket",
      "createdAt": "2021-10-22T12:42:11.792000-07:00",
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "MyExampleBucket",
      "operationType": "UpdateBucket",
      "status": "Succeeded",
      "statusChangedAt": "2021-10-22T12:42:11.792000-07:00",
      "errorCode": "",
      "errorDetails": ""
    }
  ]
}
```

Disabling access logging using the AWS CLI

Complete the following procedure to disable access logging using the AWS CLI.

Note

You must install the AWS CLI and configure it for Lightsail before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window on your local computer.
2. Enter the following command to disable access logging.

```
aws lightsail update-bucket --bucket-name SourceBucketName --access-log-config  
{"\"enabled\": false}"
```

In the command, replace *SourceBucketName* with the name of the source bucket for which to disable access logging.

Example

```
aws lightsail update-bucket --bucket-name MyExampleBucket --access-log-config  
{"\"enabled\": false}"
```

You should see a result similar to the following example after running the command.

```

aws lightsail update-bucket --bucket-name MyExampleBucket --access-log-config "{\"enabled\": false}"
{
  "bucket": {
    "resourceType": "Bucket",
    "accessRules": {
      "getObject": "private",
      "allowPublicOverrides": false
    },
    "arn": "arn:aws:lightsail:us-west-2:123456789012:bucket:MyExampleBucket",
    "bundleId": "large_1_0",
    "createdAt": "2021-06-29T08:12:39.163000-07:00",
    "url": "https://lightsail-us-west-2-123456789012.s3.amazonaws.com",
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "name": "MyExampleBucket",
    "supportCode": "lightsail-us-west-2-123456789012",
    "tags": [],
    "objectVersioning": "Suspended",
    "ableToUpdateBundle": true,
    "readonlyAccessAccounts": [
      "arn:aws:iam::123456789012:role/MyExampleRole"
    ],
    "state": {
      "code": "OK"
    },
    "accessLogConfig": {
      "enabled": false
    }
  },
  "operations": [
    {
      "id": "op-123456789012",
      "resourceName": "MyExampleBucket",
      "resourceType": "Bucket",
      "createdAt": "2021-10-22T13:24:36.881000-07:00",
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "lightsail-us-west-2-123456789012",
      "operationType": "UpdateBucket",
      "status": "Succeeded",
      "statusChangedAt": "2021-10-22T13:24:36.881000-07:00",
      "errorCode": "",
      "errorDetails": ""
    }
  ]
}

```

Use bucket access logs to identify requests in Amazon Lightsail

In this guide, we show you how to identify requests to a bucket using access logs. For more information, see [Bucket access logs](#).

Contents

- [Query access logs for requests using Amazon Athena](#)
- [Identify object access requests using Amazon S3 access logs](#)

Query access logs for requests using Amazon Athena

You can use Amazon Athena to query and identify requests to a bucket in access logs.

Lightsail stores access logs as objects in a Lightsail bucket. It is often easier to use a tool that can analyze the logs. Athena supports analysis of objects and can be used to query access logs.

Example

The following example shows how you can query bucket server access logs in Amazon Athena.

Note

To specify a bucket location in an Athena query, you need to format the target bucket name and target prefix where your logs are delivered as an S3 URI, as follows: `s3://DOC-EXAMPLE-BUCKET1-logs/prefix/`

1. Open the Athena console at <https://console.aws.amazon.com/athena/>.
2. In the **Query Editor**, run a command similar to the following.

```
create database bucket_access_logs_db
```

Note

It's a best practice to create the database in the same AWS Region as your S3 bucket.

3. In the **Query Editor**, run a command similar to the following to create a table schema in the database that you created in step 2. The `STRING` and `BIGINT` data type values are the access log properties. You can query these properties in Athena. For `LOCATION`, enter the bucket and prefix path as noted earlier.

```
CREATE EXTERNAL TABLE `s3_access_logs_db.mybucket_logs` (  
  `bucketowner` STRING,  
  `bucket_name` STRING,  
  `requestdatetime` STRING,  
  `remoteip` STRING,  
  `requester` STRING,  
  `requestid` STRING,  
  `operation` STRING,
```

```

`key` STRING,
`request_uri` STRING,
`httpstatus` STRING,
`errorcode` STRING,
`bytessent` BIGINT,
`objectsize` BIGINT,
`totaltime` STRING,
`turnaroundtime` STRING,
`referrer` STRING,
`useragent` STRING,
`versionid` STRING,
`hostid` STRING,
`sigv` STRING,
`ciphersuite` STRING,
`authtype` STRING,
`endpoint` STRING,
`tlsversion` STRING)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'input.regex'='([ ]*) ([ ]*) \\[(.?)\\] ([ ]*) ([ ]*) ([ ]*) ([ ]*)
([ ]*) (\\"[^"]*"|\\-|-|[0-9]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*)
(\\"[^"]*"|\\-|-|[0-9]*) ([ ]*)(?: ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*))?.*$')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://doc-example-bucket1-logs/prefix/'

```

4. In the navigation pane, under **Database**, choose your database.
5. Under **Tables**, choose **Preview** table next to your table name.

In the **Results** pane, you should see data from the server access logs, such as bucketowner, bucket, requestdatetime, and so on. This means that you successfully created the Athena table. You can now query the bucket server access logs.

Example — Show who deleted an object and when (timestamp, IP address, and IAM user)

```

SELECT RequestDateTime, RemoteIP, Requester, Key
FROM s3_access_logs_db.mybucket_logs
WHERE key = 'images/picture.jpg' AND operation like '%DELETE%';

```


Example — Show all operations that were performed by an IAM user

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE requester='arn:aws:iam::123456789123:user/user_name';
```

Example — Show all operations that were performed on an object in a specific time period

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE Key='prefix/images/picture.jpg'
      AND parse_datetime(RequestDateTime, 'dd/MMM/yyyy:HH:mm:ss Z')
      BETWEEN parse_datetime('2017-02-18:07:00:00', 'yyyy-MM-dd:HH:mm:ss')
      AND parse_datetime('2017-02-18:08:00:00', 'yyyy-MM-dd:HH:mm:ss');
```

Example — Show how much data was transferred by a specific IP address in a specific time period

```
SELECT SUM(bytessent) AS uploadTotal,
       SUM(objectsize) AS downloadTotal,
       SUM(bytessent + objectsize) AS Total
FROM s3_access_logs_db.mybucket_logs
WHERE RemoteIP='1.2.3.4'
      AND parse_datetime(RequestDateTime, 'dd/MMM/yyyy:HH:mm:ss Z')
      BETWEEN parse_datetime('2017-06-01', 'yyyy-MM-dd')
      AND parse_datetime('2017-07-01', 'yyyy-MM-dd');
```

Identify object access requests using Amazon S3 access logs

You can use queries on access logs to identify object access requests, for operations such as *GET*, *PUT*, and *DELETE*, and discover further information about those requests.

The following Amazon Athena query example shows how to get all PUT object requests for a bucket from the server access log.

Example — Show all requesters that are sending PUT object requests in a certain period

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db
WHERE Operation='REST.PUT.OBJECT' AND
```

```
parse_datetime(RequestDateTime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

The following Amazon Athena query example shows how to get all GET object requests for Amazon S3 from the server access log.

Example — Show all requesters that are sending GET object requests in a certain period

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db
WHERE Operation='REST.GET.OBJECT' AND
parse_datetime(RequestDateTime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

The following Amazon Athena query example shows how to get all anonymous requests to your S3 buckets from the server access log.

Example — Show all anonymous requesters that are making requests to a bucket in a certain period

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db.mybucket_logs
WHERE Requester IS NULL AND
parse_datetime(RequestDateTime, 'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42', 'yyyy-MM-dd:HH:mm:ss')
```

Note

- You can modify the date range to suit your needs.
- These query examples might also be useful for security monitoring. You can review the results for PutObject or GetObject calls from unexpected or unauthorized IP addresses/requesters and for identifying any anonymous requests to your buckets.
- This query only retrieves information from the time at which logging was enabled.

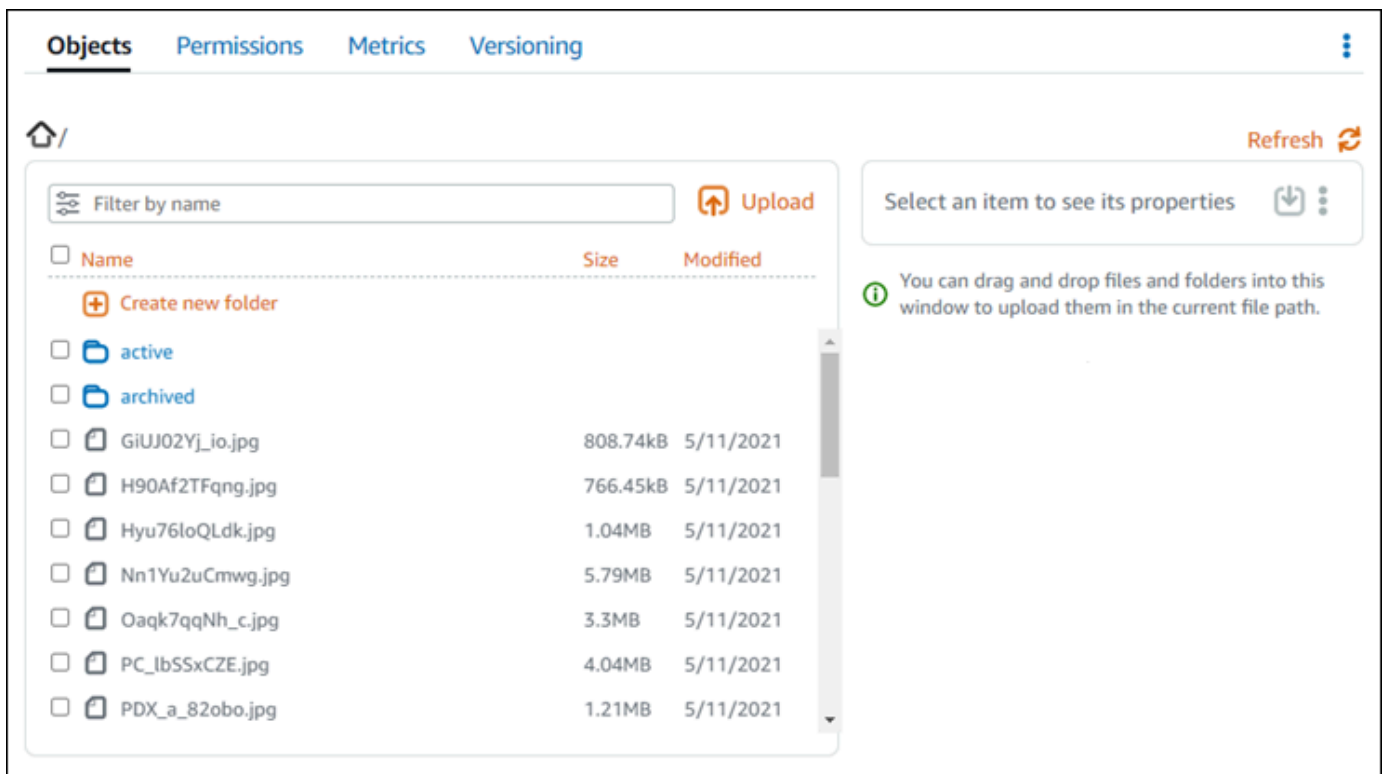
Bucket objects in Amazon Lightsail

You can view all objects stored in your bucket in the Amazon Lightsail object storage service by using the Lightsail console. You can also use the AWS Command Line Interface (AWS CLI) and AWS SDKs to list object keys in your bucket. For more information about buckets, see [Object storage](#).

Filter objects using the Lightsail console

Complete the following procedure to view objects stored in a bucket using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to view objects.
4. The **Objects browser** pane in the **Objects tab** displays the objects and folders that are stored in your bucket.



5. Browse to the location of the object for which you want to view properties.
6. Add a check mark next to the object for which you want to view properties.
7. The **Object properties** pane on the right side of the page displays information about the object.

The screenshot displays the Amazon Lightsail console's 'Objects' tab. It features a navigation bar with 'Objects', 'Permissions', 'Metrics', and 'Versioning'. A search bar is labeled 'Filter by name'. A table lists objects with columns for 'Name', 'Size', and 'Modified'. The object 'sailbot.jpg' is selected. A detailed view for 'sailbot.jpg' is shown on the right, including 'Object Size' (42.232 kB), 'Last Modified' (May 11, 2021), 'PERMISSIONS' (This object is private), 'METADATA (1% FULL)' (ContentType: image/jpeg), 'OBJECT TAGS (0/10)', and 'VERSIONS'. Red callout boxes numbered 1 through 7 point to specific UI elements: 1 (Upload icon), 2 (Actions menu), 3 (Object size), 4 (Permissions), 5 (Metadata), 6 (Object tags), and 7 (Versions).

The information displayed includes:

1. Links to view and download the object.
2. Actions menu (:) to copy or delete the object. For more information about copying and deleting objects, see [Copy or move objects in a bucket in Amazon Lightsail](#) and [Delete bucket objects](#).
3. Object size, and last modified timestamp.
4. The access permission of the individual object, which could be private or public (read-only). For more information about object permissions, see [Bucket permissions](#).
5. The metadata of the object. The content type (ContentType) key is the only metadata supported by the Lightsail object storage service at this time.
6. The object key value tags. For more information, see [Tag bucket objects](#).
7. The option to manage stored versions of the object. For more information, see [Enable and suspend object versioning in a bucket](#).

Note

When you select multiple objects, the **Object properties** pane displays only the total size of the selected objects.

View objects using the AWS CLI

Complete the following procedure to list object keys in a bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `list-objects-v2` command. For more information, see [list-objects-v2](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS Command Line Interface to work with Amazon Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter one of the following commands.
 - Enter the following command to list all object keys in your bucket.

```
aws s3api list-objects-v2 --bucket BucketName --query "Contents[].{Key: Key, Size: Size}"
```

In the command, replace *BucketName* with the name of the bucket for which you want to list all objects.

- Enter the following command to list objects that start with a specific object key name prefix.

```
aws s3api list-objects-v2 --bucket BucketName --prefix ObjectKeyNamePrefix --query "Contents[].{Key: Key, Size: Size}"
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket for which you want to list all objects.

- *ObjectKeyNamePrefix* - An object key name prefix to limit the response to keys that begin with the specified prefix.

Note

These commands use the `--query` parameter to filter the response of the `list-objects-v2` request to the key value and size of each object.

Examples:

Listing all object keys in a bucket:

```
aws s3api list-objects-v2 --bucket DOC-EXAMPLE-BUCKET --query "Contents[].{Key: Key, Size: Size}"
```

For the preceding command, you should see a result similar to the following example.

```
C:\>aws s3api list-objects-v2 --bucket DOC-EXAMPLE-BUCKET --query "Contents[].{Key: Key, Size: Size}"
[
  {
    "Key": "GiUJ02Yj_io.jpg",
    "Size": 828150
  },
  {
    "Key": "H90Af2TFqng.jpg",
    "Size": 784846
  },
  {
    "Key": "Hyu761oQLdk.jpg",
    "Size": 1086363
  },
  {
    "Key": "Nn1Yu2uCmwg.jpg",
    "Size": 6075006
  },
  {
    "Key": "Oaqk7qqNh_c.jpg",
    "Size": 3458557
  },
  {
    "Key": "PC_1bSSxCZE.jpg",
    "Size": 4239636
  },
  {
    "Key": "PDY_a_82obn.jpg"
```

Listing object keys that start with the `archived/` object key name prefix:

```
aws s3api list-objects-v2 --bucket DOC-EXAMPLE-BUCKET --prefix archived/ --query "Contents[].{Key: Key, Size: Size}"
```

For the preceding command, you should see a result similar to the following example.

```
C:\>aws s3api list-objects-v2 --bucket DOC-EXAMPLE-BUCKET --prefix archived/ --query "Contents[].{Key: Key, Size: Size}"
[
  {
    "Key": "archived/",
    "Size": 0
  },
  {
    "Key": "archived/1_CMofSPfso.jpg",
    "Size": 2561865
  },
  {
    "Key": "archived/3y1zF4hIPCg.jpg",
    "Size": 6404907
  },
  {
    "Key": "archived/5IH5WhosQE.jpg",
    "Size": 2377975
  },
  {
    "Key": "archived/sailbot.jpg",
    "Size": 43246
  }
]
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
- [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
- [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).

- 11 Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
- 12 Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
- 13 Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
- 14 Learn how to connect your bucket to other resources. For more information, see the following tutorials.
 - [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
- 15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Topics

- [Copy and move bucket objects in Amazon Lightsail](#)
- [Delete bucket objects in Amazon Lightsail](#)
- [Download objects from a bucket in Amazon Lightsail](#)
- [Filter bucket objects in Amazon Lightsail](#)
- [Enable and suspend object versioning in Amazon Lightsail](#)
- [Restore previous versions of bucket objects in Amazon Lightsail](#)
- [Tag bucket objects in Amazon Lightsail](#)

Copy and move bucket objects in Amazon Lightsail

You can copy objects that are already stored in your bucket in the Amazon Lightsail object storage service. In this guide, we show you how to copy objects using the Lightsail console and using the AWS Command Line Interface (AWS CLI). Copy objects in your bucket to create duplicate copies of objects, rename objects, or move objects across Lightsail locations (for example, moving objects from one AWS Region to another one, in which Lightsail is available). You can copy objects across locations only using the AWS APIs, AWS SDKs, and AWS Command Line Interface (AWS CLI).

For more information about buckets, see [Object storage](#).

Restrictions for copying objects

You can create a copy of an object that is up to 2 GB in size by using the Lightsail console. You can create a copy of an object that is up to 5 GB in size with a single copy object action by using the AWS Command Line Interface (AWS CLI), AWS APIs, and AWS SDKs. To copy an object that is greater than 5 GB in size, you must use the multipart upload action of the AWS CLI, AWS APIs, and AWS SDKs. For more information, see [Upload files to a bucket using multipart upload](#).

Copy objects using the Lightsail console

Complete the following procedure to copy an object stored in a bucket using the Lightsail console. To move an object in a bucket, you should copy it to the new location, and delete the original object.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to copy an object.
4. In the **Objects** tab, use the **Objects browser pane** to browse to the location of the object that you want to copy.
5. Add a check mark next to the object that you want to copy.
6. In the **Object information** pane, choose the actions (:) menu, and then choose **Copy to**.
7. In the **Select destination** pane that appears, browse to the location in the bucket where you want to copy the selected object. You can also create a new path by entering folder names into the **Destination path** text box.
8. Choose **Copy** to copy the object to the selected or specified destination. Otherwise, choose **No, cancel**.

A **Copy complete** message is displayed when the object is successfully copied. You should delete the original object if your intent was to move the object. For more information, see [Delete bucket objects](#).

Copy objects using the AWS CLI

Complete the following procedure to copy objects in a bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `copy-object` command. For more information, see [copy-object](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to copy an object in your bucket.

```
aws s3api copy-object --copy-source SourceBucketNameAndObjectKey --  
key DestinationObjectKey --bucket DestinationBucketName --acl bucket-owner-full-  
control
```

In the command, replace the following example text with your own:

- *SourceBucketNameAndObjectKey* - The name of the bucket in which the source object currently exists, and the full object key of the object to be copied. For example, to copy the object `images/sailbot.jpg` from the bucket `DOC-EXAMPLE-BUCKET`, specify `DOC-EXAMPLE-BUCKET/images/sailbot.jpg`.
- *DestinationObjectKey* - The full object key of the new object copy.
- *DestinationBucket* - The name of the destination bucket.

Examples:

- Copying an object in a bucket to the same bucket:

```
aws s3api copy-object --copy-source DOC-EXAMPLE-BUCKET/images/sailbot.jpg --  
key media/sailbot.jpg --bucket DOC-EXAMPLE-BUCKET --acl bucket-owner-full-control
```

- Copying an object from one bucket to another bucket:

```
aws s3api copy-object --copy-source DOC-EXAMPLE-BUCKET-1/images/sailbot.jpg --  
key images/sailbot.jpg --bucket DOC-EXAMPLE-BUCKET-2 --acl bucket-owner-full-  
control
```

You should see a result similar to the following example:

```
C:\>aws s3api copy-object --copy-source DOC-EXAMPLE-BUCKET/images/sailbot.jpg --key images/archived/sailbot.jpg --bucket DOC-EXAMPLE-BUCKET
{
  "ServerSideEncryption": "AES256",
  "CopyObjectResult": {
    "ETag": "\"694d34example91d92d64f342aa234c3\"",
    "LastModified": "2021-05-10T05:35:42+00:00"
  }
}
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)

- [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
 7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
 8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
 9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
 10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
 11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
 12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
 13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
 14. Learn how to connect your bucket to other resources. For more information, see the following tutorials.
 - [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)

Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Delete bucket objects in Amazon Lightsail

You can delete objects from your bucket in the Amazon Lightsail object storage service. To free-up storage space, delete objects that you no longer need. For example, if you're collecting log files, it's a good idea to delete them when you don't need them anymore.

For more information about buckets, see [Object storage](#).

Contents

- [Delete objects from a version-enabled bucket](#)
- [Delete objects using the Lightsail console](#)
- [Delete object versions using the Lightsail console](#)
- [Delete a single object or object version using the AWS CLI](#)
- [Delete multiple objects or object versions using the AWS CLI](#)

Delete objects from a version-enabled bucket

If versioning is enabled on your bucket, multiple versions of the same object can exist in it. You can delete any version of an object using the Lightsail console, AWS CLI, AWS APIs, or AWS SDKs. However, you should consider the following options.

Delete objects and object versions using the Lightsail console

When you delete the current version of an object in the **Objects browser pane** of the **Objects** tab in the Lightsail console, this also deletes all previous versions of the object. To delete a specific version of an object, you must do so from the **Manage versions** pane. If you use the **Manage versions** pane to delete the current version of an object, then the most recent previous version is restored as the current version. For more information, see [Delete object versions using the Lightsail console](#) later in this guide.

Delete objects and object versions using the Lightsail API, AWS CLI, or AWS SDKs

To delete a single object and all of its stored versions, specify only the object's key in your delete request. To delete a specific version of an object, specify both the object key and also a version ID.

For more information, see [Delete a single object or object version using the AWS CLI](#) later in this guide.

Delete objects using the Lightsail console

Complete the following procedure to delete an object, including its stored previous versions, using the Lightsail console. You can delete only one object at a time using the Lightsail console. Use the AWS CLI to delete multiple objects at once. For more information, see [Delete multiple objects or object versions using the AWS CLI](#) later in this guide.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to delete objects.
4. Use the **Objects browser** pane in the **Objects** tab to browse to the location of the object that you want to delete.
5. Add a check mark next to the object that you want to delete.
6. In the **Object information** pane, choose the actions (:) menu, and then choose **Delete**.
7. In the confirmation pane that appears, confirm that you want to permanently delete the object by choosing **Yes, delete**.

If you delete the only object in the folder that you're in, this also deletes the folder. This happens because the folder is part of the object key name, and deleting the object also deletes the preceding folders when no other objects in the bucket share the same object prefix. For more information, see [Key names for object storage buckets](#).

Delete object versions using the Lightsail console

Complete the following procedure to delete stored versions of an object. This is only possible for version-enabled buckets. For more information, see [Enable and suspend object versioning in a bucket](#).

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to delete objects.
4. Use the **Objects browser** pane to browse to the location of the object that you want to delete.
5. Add a check mark next to the object for which you want to delete stored previous versions.

6. Choose **Manage** in the **Versions** section of the **Object information** pane, and then choose **Manage**.
7. In the **Manage stored object versions** pane that appears, add a check mark next to the versions of the object that you want to delete.

You can also choose to delete the current version of an object.

8. Choose **Delete selected** to delete the selected versions.

If you delete:

- The current version of an object - The most recent previous version of the object is restored as the current version.
- The only version of an object - The object is deleted from the bucket. If the version you deleted is the only object in the current folder, then the folder is deleted also. This happens because the folder is part of the object key name, and deleting the object also deletes the preceding folders when no other objects in the bucket share the same object key prefix. For more information, see [Enable and suspend object versioning in a bucket](#).

Delete a single object or object version using the AWS CLI

Complete the following procedure to delete a single object or object version in your bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `delete-object` command. For more information, see [delete-object](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS Command Line Interface to work with Amazon Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to delete an object or an object version in your bucket.

To delete an object:

```
aws s3api delete-object --bucket BucketName --key ObjectKey
```


To delete an object version:

Note

Deleting object versions is only possible for version-enabled buckets. For more information, see [Enable and suspend object versioning in a bucket](#).

```
aws s3api delete-object --bucket BucketName --key ObjectKey --version-id VersionID
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket from which you want to delete an object.
- *ObjectKey* - The full object key of the object you want to delete.
- *VersionID* - The ID of the object version you want to delete.

Examples:

Deleting an object:

```
aws s3api delete-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg
```

Deleting an object version:

```
aws s3api delete-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg --  
version-id YF0YMB1Uvexample00712vJi9hRz4ujX
```

You should see a result similar to the following example:

```
C:\Users\latino>aws s3api delete-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg --version-id YF0YMB1Uvexample00712vJi9hRz4ujX  
{  
  "VersionId": "YF0YMBexampleY7P00712vJi9hRz4ujX"  
}
```

Delete multiple objects or object versions using the AWS CLI

Complete the following procedure to delete multiple objects in your bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `delete-objects` command. For more information, see [delete-objects](#) in the AWS CLI Command Reference.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS Command Line Interface to work with Amazon Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to delete multiple objects or multiple object versions in your bucket.

```
aws s3api delete-objects --bucket BucketName --delete file:///LocalDirectory
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket from which you want to delete multiple objects or multiple object versions.
- *LocalDirectory* - The directory path on your computer of the .json document that specifies the objects or versions to delete. The .json document can be formatted as follows.

To delete objects, enter the following text in the .json file and replace *ObjectKey* with the object key of the objects you want to delete.

```
{
  "Objects": [
    {
      "Key": "ObjectKey1"
    },
    {
      "Key": "ObjectKey2"
    }
  ],
  "Quiet": false
}
```

```
}
```

To delete object versions, enter the following text in the .json file. Replace *ObjectKey* and *VersionID* with the object key and IDs of the object versions that you want to delete.

Note

Deleting object versions is only possible for version-enabled buckets. For more information, see [Enable and suspend object versioning in a bucket](#).

```
{
  "Objects": [
    {
      "Key": "ObjectKey1",
      "VersionId": "VersionID1"
    },
    {
      "Key": "ObjectKey2",
      "VersionId": "VersionID2"
    }
  ],
  "Quiet": false
}
```

Examples:

- On a Linux or Unix computer:

```
aws s3api delete-objects --bucket DOC-EXAMPLE-BUCKET --delete file:///home/user/
Documents/delete-objects.json
```

- On a Windows computer:

```
aws s3api delete-objects --bucket DOC-EXAMPLE-BUCKET --delete file:///C:\Users
\user\Documents\delete-objects.json
```

You should see a result similar to the following example:

```
C:\>aws s3api delete-objects --bucket DOC-EXAMPLE-BUCKET --delete file:///C:/Users/user/Documents/delete-objects.json
{
  "Deleted": [
    {
      "Key": "images/sailbot.jpg",
      "VersionId": "26sqexampleztrIT6TsGhMMz0FxQAEw."
    },
    {
      "Key": "images/sailbot.jpg",
      "VersionId": "QwDrexampleDJxJtZC1CrExbpN1EC504"
    }
  ]
}
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.

- [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
 7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
 8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
 9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
 10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
 11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
 12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
 13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
 14. Learn how to connect your bucket to other resources. For more information, see the following tutorials.
 - [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)

- [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)

15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

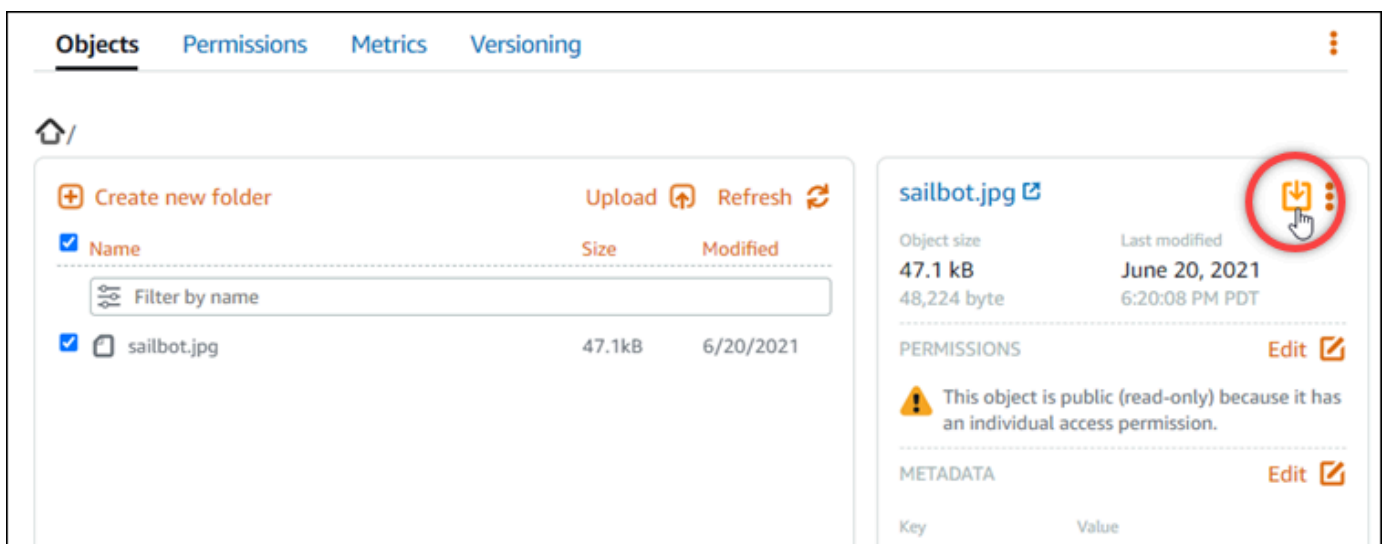
Download objects from a bucket in Amazon Lightsail

You can download objects from buckets that you have access to or that are public (read-only) in the Amazon Lightsail object storage service. You can download a single object at a time using the Lightsail console. To download multiple objects in one request, use the AWS Command Line Interface (AWS CLI), AWS SDKs, or REST API. In this guide, we show you how to download objects using the Lightsail console and AWS CLI. For more information about buckets, see [Object storage](#).

Download objects using the Lightsail console

Complete the following procedure to download objects from a bucket using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket from which you want to download a file.
4. In the **Objects** tab, use the **Objects browser pane** to browse to the location of the object that you want to download.
5. Add a check mark next to the object that you want to download.
6. In the **Object information** pane, choose the download icon.



Depending on the configuration of your browser, the file that you chose is either displayed on the page or is downloaded to your computer. If the file is displayed on the page, you can right-click it and choose **Save as** to save it to your computer.

Download objects using the AWS CLI

Complete the following procedure to download objects from a bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `get-object` command. For more information, see [get-object](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS Command Line Interface to work with Amazon Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to download an object from your bucket.

```
aws s3api get-object --bucket BucketName --key ObjectKey LocalFilePath
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket from which you want to download an object.
- *ObjectKey* - The full object key of the object you want to download.
- *LocalFilePath* - The full file path on your computer where you want to save the downloaded file.

Example:

```
aws s3api get-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg C:\Users\user\Pictures\sailbot.jpg
```

You should see a result similar to the following example:

```
C:\>aws s3api get-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg C:\Users\user\Pictures\sailbot.jpg
{
  "AcceptRanges": "bytes",
  "LastModified": "2021-05-10T05:09:31+00:00",
  "ContentLength": 48224,
  "ETag": "\"694d34example91d92d64f342aa234c3\"",
  "ContentType": "binary/octet-stream",
  "ServerSideEncryption": "AES256",
  "Metadata": {}
}
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)

- [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
 7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
 8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
 9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
 10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
 11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
 12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
 13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
 14. Learn how to connect your bucket to other resources. For more information, see the following tutorials.
 - [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)

- [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)

15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Filter bucket objects in Amazon Lightsail

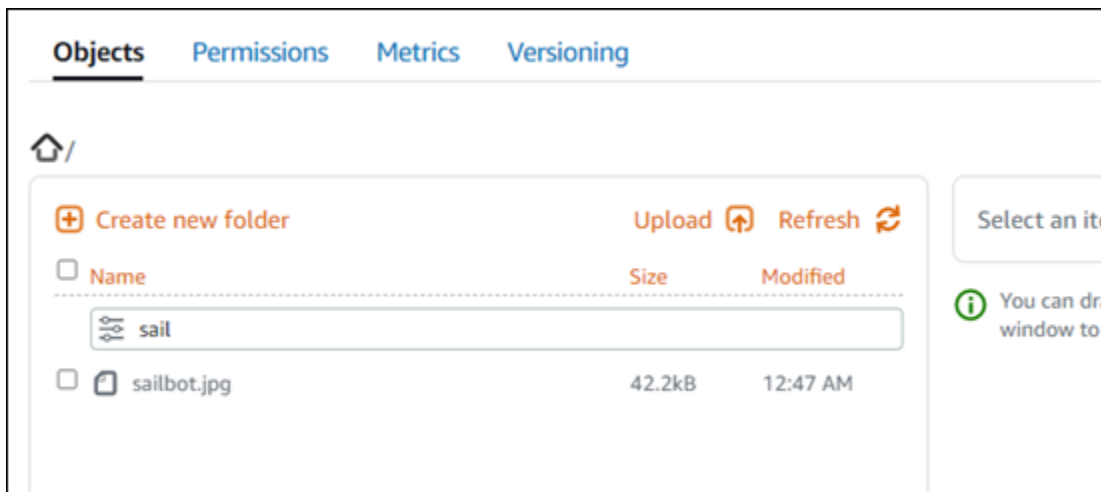
You can use filtering to find objects in your bucket in the Amazon Lightsail object storage service. In this guide, we show you how to filter objects using the Lightsail console, and the AWS Command Line Interface (AWS CLI). For more information about buckets, see [Object storage](#).

Filter objects using the Lightsail console

Complete the following procedure to filter objects in a bucket using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to find objects.
4. In the **Objects** tab, type an object prefix in the **Filter by name** text box.

The list of objects in the folder that you're currently viewing are filtered to match the text you enter. The following example shows that if you enter `sail`, the list of objects on the page are filtered to display only those that start with `sail`.



To filter the list of objects in a different folder, navigate to that folder. Then, enter the object prefix into the **Filter by name** text box there.

Filter objects using the AWS CLI

Complete the following procedure to filter objects in a bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `list-objects-v2` command. For more information, see [list-objects-v2](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS Command Line Interface to work with Amazon Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to list objects that start with a specific object key name prefix.

```
aws s3api list-objects-v2 --bucket BucketName --prefix ObjectKeyNamePrefix --query "Contents[].{Key: Key, Size: Size}"
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket for which you want to list all objects.
- *ObjectKeyNamePrefix* - An object key name prefix to limit the response to keys that begin with the specified prefix.

Note

This command uses the `--query` parameter to filter the response of the `list-objects-v2` request to the key value and size of each object.

Example:

```
aws s3api list-objects-v2 --bucket DOC-EXAMPLE-BUCKET --prefix archived/ --query "Contents[].{Key: Key, Size: Size}"
```

You should see a result similar to the following example.

```
C:\>aws s3api list-objects-v2 --bucket DOC-EXAMPLE-BUCKET --prefix archived/ --query "Contents[].[Key: Key, Size: Size]"
[
  {
    "Key": "archived/",
    "Size": 0
  },
  {
    "Key": "archived/1_CMoFsPfso.jpg",
    "Size": 2561865
  },
  {
    "Key": "archived/3y1zF4hIPCg.jpg",
    "Size": 6404907
  },
  {
    "Key": "archived/5IHZ5WhosQE.jpg",
    "Size": 2377975
  },
  {
    "Key": "archived/sailbot.jpg",
    "Size": 43246
  }
]
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
- [Configuring bucket access permissions in Amazon Lightsail](#)
- [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
- [Creating access keys for a bucket in Amazon Lightsail](#)

- [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
 6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
 7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
 8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
 9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
 10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
 11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
 12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).

- 13 Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
- 14 Learn how to connect your bucket to other resources. For more information, see the following tutorials.
 - [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
- 15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Enable and suspend object versioning in Amazon Lightsail

Versioning in Amazon Lightsail object storage service is a means of keeping multiple variants of an object in the same bucket. You can use the versioning feature to preserve, retrieve, and restore every version of every object stored in your buckets. With versioning, you can recover more easily from both unintended user actions and application failures. When you enable versioning for a bucket, if the Lightsail object storage service receives multiple write requests for the same object simultaneously, it stores all of those objects. Versioning is disabled by default on buckets in the Lightsail object storage service, so you must explicitly enable it. For more information about buckets, see [Object storage](#).

Important

When you enable or suspend versioning on a bucket that has the **Individual objects can be made public (read-only)** access permission configured, the permission resets to **All objects are private**. If you want to continue having the option to make individual objects public, you must manually change the bucket access permission back to **Individual objects can be made public (read-only)**. For more information, see [Configure bucket access permissions](#).

Version disabled, enabled, and suspended buckets

Bucket versioning can be in one of three states in the Lightsail console:

- Disabled (NeverEnabled in the API and SDKs)
- Enabled (Enabled in the API and SDKs)

- Suspended (Suspended in the API and SDKs)

After you enable versioning in a bucket, it cannot return to a disabled state. But you can suspend versioning. You enable and suspend versioning at the bucket level.

The versioning state applies to all (not some) of the objects in that bucket. When you enable versioning in a bucket, all new objects are versioned and given a unique version ID. Objects that already exist in the bucket when versioning is enabled are always versioned going forward. They are given a unique version ID when they are modified by future requests.

Version IDs

If you enable versioning for a bucket, the Lightsail object storage service automatically generates a unique version ID for the object that is being stored. For example, in one bucket you can have two objects with the same key but different version IDs, such as `photo.gif` (version 111111) and `photo.gif` (version 121212).



Version IDs cannot be edited. They are Unicode, UTF-8 encoded, URL-ready, opaque strings that are no more than 1,024 bytes long. The following is an example of a version ID:

```
3sL4kqtJlcpXroDTdMj+imSpXd3dIb1rHY+MTRCxf3vjVBH40N1r8X8gdRQBpUMLUo
```

Enable or suspend object versioning using the Lightsail console

Complete the following procedure to enable or suspend object versioning using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to enable or suspend versioning.
4. Choose the Versioning tab.

5. Complete one of the following actions depending on the current versioning state of your bucket:
 - If versioning is currently suspended or has not been enabled, choose the toggle under the **Object versioning** section of the page to enable versioning.
 - If versioning is currently enabled, choose the toggle under the **Object versioning** section of the page to suspend versioning.

Enable or suspend object versioning using the AWS CLI

Complete the following procedure to enable or suspend object versioning using the AWS Command Line Interface (AWS CLI). You do this by using the `update-bucket` command. For more information, see [update-bucket](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to enable or suspend object versioning.

```
aws lightsail update-bucket --bucket-name BucketName --versioning VersioningState
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket for which you want to enable object versioning.
- *VersioningState* - One of the following:
 - Enabled - Enables object versioning.
 - Suspended - Suspends object versioning if it was previously enabled.

Example:

```
aws lightsail update-bucket --bucket-name DOC-EXAMPLE-BUCKET --versioning Enabled
```


You should see a result similar to the following example:

```
C:\>aws lightsail update-bucket --bucket-name DOC-EXAMPLE-BUCKET --versioning Enabled
{
  "bucket": {
    "resourceType": "Bucket",
    "accessRules": {
      "getObject": "private",
      "allowPublicOverrides": false
    },
    "arn": "arn:aws:lightsail:us-west-2:1example7491:Bucket/f067383e-ee41-4485-b934-example2e2fd",
    "bundleId": "small_1_0",
    "createdAt": "2021-06-29T08:12:39.163000-07:00",
    "url": "https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com",
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "name": "DOC-EXAMPLE-BUCKET",
    "supportCode": "621291663362/DOC-EXAMPLE-BUCKET/small_1_0",
    "tags": [],
    "objectVersioning": "Enabled",
    "ableToUpdateBundle": true
  },
  "operations": [
    {
      "id": "0d53d290-f4b2-43f0-89d2-example43448",
      "resourceName": "DOC-EXAMPLE-BUCKET",
      "resourceType": "Bucket",
      "createdAt": "2021-06-29T08:29:56.241000-07:00",
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "6example3362/DOC-EXAMPLE-BUCKET/small_1_0",
      "operationType": "UpdateBucket",
      "status": "Succeeded",
      "statusChangedAt": "2021-06-29T08:29:56.241000-07:00",
      "errorCode": "",
      "errorDetails": ""
    }
  ]
}
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).

3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
 6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
 7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
 8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)

- [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
14. Learn how to connect your bucket to other resources. For more information, see the following tutorials.
- [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
15. Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Restore previous versions of bucket objects in Amazon Lightsail

If your bucket in the Amazon Lightsail object storage service is version-enabled, then you can restore previous versions of an object. Restore a previous version of an object recover from unintended user actions or application failures.

You can restore a previous version of an object using the Lightsail console. You can also use the AWS Command Line Interface (AWS CLI) and AWS SDKs restore a previous version of an object. To do this, copy a specific version of the object into the same bucket, and use the same object key name. This replaces the current version with the previous version, making the previous version the

current version. For more information about versioning, see [Enable and suspend bucket object versioning](#). For more information about buckets, see [Object storage](#).

Restore a previous version of an object using the Lightsail console

Complete the following procedure to restore a previous version of an object using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to restore a previous version of an object.
4. Use the **Objects browser** pane in the **Objects** tab to browse to the location of the object.
5. Add a check mark next to the object for which you want to restore a previous version.
6. Choose **Manage** under the Versions section of the **Object information** pane.
7. Choose **Restore**.
8. In the **Restore object from a stored version** pane that appears, choose the version of the object that you want to restore.
9. Choose **Continue**.
10. In the confirmation prompt that appears, choose **Yes, restore** to restore the object version. Otherwise, choose **No, cancel**.

Restore a previous version of an object using the AWS CLI

Complete the following procedure to restore a previous version of an object the AWS Command Line Interface (AWS CLI). You do this by using the `copy-object` command. You must copy the previous version of the object into the same bucket, using the same object key. For more information, see [copy-object](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS Command Line Interface to work with Amazon Lightsail](#).

1. Open a Command Prompt or Terminal window.

2. Enter the following command to restore a previous version of an object.

```
aws s3api copy-object --copy-source "BucketName/ObjectName?versionId=VersionId" --  
key ObjectKey --bucket BucketName
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket for which you want to restore a previous version of an object. You must specify the same bucket name for the `--copy-source` and `--bucket` parameters.
- *ObjectKey* - The name of the object to restore. You must specify the same object key name for the `--copy-source` and `--key` parameters.
- *VersionId* - The ID of the previous object version that you want to restore to the current version. Use the `list-object-versions` command to get a list of version IDs for objects in your bucket.

Example:

```
aws s3api copy-object --copy-source "DOC-EXAMPLE-BUCKET/sailbot.jpg?  
versionId=GQWEexample87Md18Q_DKdVTiVMi_VyU" -key sailbot.jpg --bucket DOC-EXAMPLE-  
BUCKET
```

You should see a result similar to the following example:

```
C:\>aws s3api copy-object --copy-source "DOC-EXAMPLE-BUCKET/sailbot.jpg?versionId=GQWEexample87Md18Q_DKdVTiVMi_VyU"  
--key sailbot.jpg --bucket DOC-EXAMPLE-BUCKET  
{  
  "CopySourceVersionId": "GQWEcouyrfexampleQ_DKdVTiVMi_VyU",  
  "VersionId": "hjl8ankzI1xcXYexampleDvvqMXSLoi",  
  "ServerSideEncryption": "AES256",  
  "CopyObjectResult": {  
    "ETag": "\"dc5afd388fb3example20cda3fe41c54\"",  
    "LastModified": "2021-05-16T06:45:35+00:00"  
  }  
}
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).

2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
 6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
 7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
 8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)

- [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
- 10 After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
- 11 Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
- 12 Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
- 13 Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
- 14 Learn how to connect your bucket to other resources. For more information, see the following tutorials.
- [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
- 15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Tag bucket objects in Amazon Lightsail

Tag objects in your bucket to categorize them by purpose, owner, environment, or other criteria. Tags can be added to objects when you upload them, or after they are uploaded. For more information about buckets, see [Object storage](#).

Add and delete tags for objects using the Lightsail console

Complete the following procedure to add or delete tags from objects in a bucket using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to tag objects.
4. Use the **Objects browser** pane in the **Objects** tab to browse to the location of the object.
5. Add a check mark next to the object for which you want to add or delete a tag.
6. In the object information pane, choose one of the following options under the **Object tags** section:
 - **Add** or **Edit** (if tags have already been added). Enter a key into the Key text box, and a value into the **Value** text box. Then, choose **Save** to add the tag. Otherwise, choose **Cancel**.
 - **Edit**, and then choose the **X** next to the key-value tag that you want to delete. Choose **Save** when you're done to delete the tag, or choose **Cancel** to not delete it.

Add and delete tags for objects using the AWS CLI

Complete the following procedure to add tags to objects or delete tags from objects using the AWS Command Line Interface (AWS CLI). You do this by using the `put-object-tagging` and `delete-object-tagging` commands. For more information, see [put-object-tagging](#) and [delete-object-tagging](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter one of the following commands:
 - To add a tag to an object:


```
aws s3api put-object-tagging --bucket BucketName --key ObjectKey --tagging
"{\"TagSet\": [{ \"Key\": \"KeyTag\", \"Value\": \"ValueTag\" } ]}"
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket that contains the object you want to tag.
- *ObjectKey* - The full object key of the object you want to tag.
- *KeyTag* - The key value of your tag.
- *ValueTag* - The value of your tag.
- To add a tag to an object:

```
aws s3api put-object-tagging --bucket BucketName --key ObjectKey --tagging
"{\"TagSet\": [{ \"Key\": \"KeyTag1\", \"Value\": \"ValueTag1\" }, { \"Key\":
\"KeyTag2\", \"Value\": \"ValueTag2\" } ]}"
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket that contains the object you want to tag.
- *ObjectKey* - The full object key of the object you want to tag.
- *KeyTag1* - The key value of your first tag.
- *ValueTag1* - The value of your first tag.
- *KeyTag2* - The key value of your second tag.
- *ValueTag2* - The value of your second tag.
- To delete all tags from an object:

```
aws s3api delete-object-tagging --bucket BucketName --key ObjectKey
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket that contains the object for which you want to delete all tags.
- *ObjectKey* - The full object key of the object you want to tag.

Example:

```
aws s3api delete-object --bucket DOC-EXAMPLE-BUCKET --key nptLmg6jqDo.jpg --tagging
"{\"TagSet\": [{ \"Key\": \"Importance\", \"Value\": \"High\" } ]}"
```

You should see a result similar to the following example:

```
C:\>aws s3api put-object-tagging --bucket DOC-EXAMPLE-BUCKET --key nptLmg6jqDo.jpg
--tagging "{\"TagSet\": [{ \"Key\": \"Importance\", \"Value\": \"High\" } ]}"
{
  "VersionId": "9nL2d41NuZdhdk4HS3kZIw0xJeS1kCkm"
}
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
- [Configuring bucket access permissions in Amazon Lightsail](#)
- [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
- [Creating access keys for a bucket in Amazon Lightsail](#)
- [Configuring resource access for a bucket in Amazon Lightsail](#)

- [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
 6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
 7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
 8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
 9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
 10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
 11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
 12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
 13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).

14 Learn how to connect your bucket to other resources. For more information, see the following tutorials.

- [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
- [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)

15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Configure resource access for a Lightsail bucket

Attach an Amazon Lightsail instance to a Lightsail bucket to give it full programmatic access to the bucket and its objects. When you attach instances to buckets, you don't have to manage credentials like access keys. The instances and buckets that you attach must be in the same AWS Region. You cannot attach instances to buckets that are in a different Region.

Resource access is ideal if you're configuring software or a plugin on your instance to upload files directly to your bucket. For example, if you want to configure a WordPress instance to store media files on a bucket. For more information, see [Tutorial: Connect a bucket to your WordPress instance](#).

For more information about permission options, see [Bucket permissions](#). For more information about security best practices, see [Security Best Practices for object storage](#). For more information about buckets, see [Object storage](#).

Configure resource access for a bucket

Complete the following procedure to configure resource access for a bucket.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to configure resource access.
4. Choose the **Permissions** tab.

The **Resource access** section of the page displays the instances currently attached to the bucket, if any.

5. Choose **Attach instance** to attach an instance to the bucket.
6. In the **Select an instance** dropdown menu, select the instance that you want to attach to the bucket.

Note

You can attach instances that are in a running or stopped state only. Additionally, you can attach only instances that are in the same AWS Region as the bucket.

7. Choose **Attach** to attach the instance. Otherwise, choose **Cancel**.

The instance has full access to the bucket and its objects after it's attached. You can configure software or a plugin on your instance to programmatically upload and access files on your bucket. For example, if you want to configure a WordPress instance to store media files on a bucket. For more information, see [Tutorial: Connect a bucket to your WordPress instance](#).

Change the plan of your Lightsail bucket

In the Amazon Lightsail object storage service, a bucket's storage plan specifies its monthly cost, storage space quota, and data transfer quota. You can update your bucket's storage plan only one time within a monthly AWS billing cycle. When you change your bucket's storage plan, the storage space and network transfer quotas are reset. However, the excess storage space and data transfer charges you might have incurred from using the previous storage plan are not covered.

Update your bucket's storage plan if it's consistently going over its storage space or data transfer quota, or if your bucket's usage is consistently in the lower range of these quotas. Because your bucket might experience unpredictable usage fluctuations, we strongly recommend that you update your bucket's storage plan only as a long-term strategy, instead of as a short-term, monthly cost-cutting measure. Choose a storage plan that will provide your bucket with an ample storage space and data transfer quota for a long time to come.

For more information about buckets, see [Object storage](#).

Change your bucket's storage plan using the Lightsail console

Complete the following procedure to change your bucket's storage plan using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to change the plan.
4. Choose the **Metrics** tab in the bucket management page.

5. Choose **Change storage plan**.
6. In the confirmation prompt that appears, choose **Yes, change** to continue to change your bucket storage plan. Otherwise, choose **No, cancel**.
7. Choose the plan that you want to use, and then choose **Select plan**.
8. In the confirmation prompt that appears, choose **Yes, apply** to apply the change to your bucket, or choose **No, go back** to not apply it.

Change your bucket's storage plan using the AWS CLI

Complete the following procedure to change the plan of your bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `update-bucket-bundle` command. Note that a bucket storage plan is referred to as a bucket bundle in the API. For more information, see [update-bucket-bundle](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to change the plan of your bucket.

```
aws lightsail update-bucket-bundle --bucket-name BucketName --bundle-id BundleID
```

In the command, replace the following example text with your own:

- *BucketName* - The name of the bucket for which you want to update the storage plan.
- *BundleID* - The ID of the new bucket bundle you want to apply to the bucket. Use the `get-bucket-bundles` command to see a list of available bucket bundles and their IDs. For more information, see [get-bucket-bundles](#) in the *AWS CLI Command Reference*.

Example:

```
aws lightsail update-bucket-bundle --bucket-name DOC-EXAMPLE-BUCKET --bundle-id medium_1_0
```

You should see a result similar to the following example:

```
C:\>aws lightsail update-bucket-bundle --bucket-name DOC-EXAMPLE-BUCKET --bundle-id medium_1_0
{
  "operations": [
    {
      "id": "8example-8176-48bd-b1da-exampleb8404",
      "resourceName": "DOC-EXAMPLE-BUCKET",
      "resourceType": "Bucket",
      "createdAt": "2021-06-30T12:05:57.362000-07:00",
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "62example362/DOC-EXAMPLE-BUCKET/medium_1_0",
      "operationType": "UpdateBucketBundle",
      "status": "Succeeded",
      "statusChangedAt": "2021-06-30T12:05:57.362000-07:00",
      "errorCode": "",
      "errorDetails": ""
    }
  ]
}
```

Configure Lightsail bucket access permissions

Use bucket access permissions to control public (unauthenticated) read-only access to objects in a bucket. You can make a bucket private or public (read-only). You can also make a bucket private, while having the option to make individual objects public (read-only).

Important

When you make a bucket public (read-only), you make all objects in the bucket readable by anyone on the internet through the bucket's URL (for example, <https://DOC-EXAMPLE-BUCKET.us-east-1.amazonaws.com/media/sailbot.jpg>). Don't make a bucket public (read-only) if you don't want anyone on the internet to have access to your objects.

For more information about permission options, see [Bucket permissions](#). For more information about security best practices, see [Security Best Practices for object storage](#). For more information about buckets, see [Object storage](#).

Important

Lightsail object storage resources take into account both Lightsail bucket access permissions and Amazon S3 account-level block public access configurations when allowing or denying public access. For more information, see [Block public access for buckets](#).

Configure bucket access permissions

Complete the following procedure to configure access permissions for a bucket.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to configure access permissions.
4. Choose the **Permissions** tab.

The **Bucket access permissions** section of the page displays the currently configured access permission for the bucket.

5. Choose **Change permission** to change the bucket access permissions.
6. Choose one of the following options:
 - **All objects are private** – All objects in the bucket are readable only by you or anyone you give access to.
 - **Individual objects can be made public (read-only)** – Objects in the bucket are readable only by you or anyone you give access to, unless you specify an individual object to be public (read-only). For more information about individual object access permissions, see [Configure access permissions for individual objects in a bucket](#).

We recommend that you select the **Individual objects can be made public (read-only)** option only if you have a specific need to do so, such as making only some of the objects in your bucket public while keeping all other objects private. For example, some WordPress plugins require that your bucket allows individual objects to be made public. For more

information, see [Tutorial: Connect a bucket to your WordPress instance](#) and [Tutorial: Use a bucket with a content delivery network distribution](#).

- **All objects are public (read-only)** – All objects in the bucket are readable by anyone on the internet.

Important

When you make a bucket public (read-only), you make all objects in the bucket readable by anyone on the internet through the bucket's URL (for example, `https://DOC-EXAMPLE-BUCKET.us-east-1.amazonaws.com/media/sailbot.jpg`). Don't make a bucket public (read-only) if you don't want anyone on the internet to have access to your objects.

7. Choose **Save** to save the change. Otherwise, choose **Cancel**.

The following changes are implemented depending on which bucket access permission you change to:

- **All objects are private** - All objects in the bucket become private even if they were previously configured with a **Public (read-only)** individual object access permission.
- **Individual objects can be made public (read-only)** - Objects that were previously configured with a **Public (read-only)** individual object access permission become public. You can now configure individual object access permissions for objects.
- **All objects are public (read-only)** - All objects in the bucket become public (read-only) even if they were previously configured with a **Private** individual object access permission.

For more information about individual object access permissions, see [Configure access permissions for individual objects in a bucket](#).

Configure cross-account access for a Lightsail bucket

Use cross-account access to grant read-only access to all objects in a bucket for other AWS accounts and their users. Cross-account access is ideal if you want to share objects with another AWS account. When you grant cross-account access to another AWS account, users in that account have read-only access to objects in a bucket through the URL of the bucket and objects (for example, `https://DOC-EXAMPLE-BUCKET.us-east-1.amazonaws.com/media/sailbot.jpg`). You can give bucket access to a maximum of 10 AWS accounts.

For more information about permission options, see [Bucket permissions](#). For more information about security best practices, see [Security Best Practices for object storage](#). For more information about buckets, see [Object storage](#).

Configure cross-account access for a bucket

Complete the following procedure to configure cross-account access for a bucket.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to configure cross-account access.
4. Choose the **Permissions** tab.

The **Cross-account access** section of the page displays the AWS account IDs that are currently configured to access the bucket, if any.

5. Choose **Add cross-account access** to grant access to the bucket for another AWS account.
6. Enter the ID of the AWS account for which you want to grant access in the **Account ID** text box.
7. Choose **Save** to grant access. Otherwise, choose **Cancel**.

The AWS account ID you added is listed in the **Cross-account access** section of the page. To remove cross-account access for an AWS account, choose the delete (trash can) icon next to the AWS account ID that you want to remove.

Configure access permissions for individual bucket objects in Lightsail

Use individual object access permissions to control public (unauthenticated) read-only access to individual objects in a bucket. You can make individual objects in a bucket private or public (read-only).

Important

Individual object access permissions can be configured only when the access permission of a bucket is set to **Individual objects can be made public (read-only)**. For more information about bucket permission options, see [Bucket permissions](#). For more information about buckets, see [Object storage](#).

We recommend that you configure individual object access permissions only if you have a specific need to do so, such as making only some of the objects in your bucket public while keeping all other objects private. For example, some WordPress plugins require that your bucket allows individual objects to be made public. For more information, see [Tutorial: Connect a bucket to your WordPress instance](#) and [Tutorial: Use a bucket with a content delivery network distribution](#).

For more information about permission options, see [Bucket permissions](#). For more information about security best practices, see [Security Best Practices for object storage](#). For more information about buckets, see [Object storage](#).

Configure individual object access permissions

Complete the following procedure to configure access permissions for an individual object in a bucket. For an example IAM policy that grants a user the ability to manage a bucket in Lightsail, see , [IAM policy to manage buckets](#).

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to configure access permissions for an individual object.
4. Choose the **Objects** tab.
5. Add a check mark next to the object for which you want to configure an access permission.

The object information pane displays the current access permissions for the object.

6. Choose **Edit** in the **Permissions** section of the object information pane to change the access permission for the object.

Note

If the edit option is not available, then the access permission of your bucket does not allow for individual object access permissions to be configured. To configure individual object access permissions, the bucket access permission must be set to **Individual objects can be made public (read-only)**. For more information, see [Configure bucket access permissions](#).

7. Choose one of the following options in the **Select a permission** dropdown menu:
 - **Private** – The object is readable only by you or anyone you give access to.

- **Public (read-only)** – The object is readable by anyone in the world.
8. Choose **Save** to save the change. Otherwise, choose **Cancel**.

The **Bucket access permission** setting of the bucket has the following effects on individual object access permissions:

- If you change the bucket access permission to **All objects are private**, all objects in the bucket become private even if they were configured with a **Public (read-only)** individual object access permission. However, individual object access permissions that were configured are retained. For example, if you change the bucket access permission back to **Individual objects can be made public (read-only)**, all objects with a **Public (read-only)** individual access permission become publicly readable again.
- If you change the bucket access permission to **All objects are public (read-only)**, all objects in the bucket become public (read-only), even if they were configured with a **Private** individual object access permission.

For more information about bucket access permissions, see [Configure bucket access permissions](#).

Upload files to a Lightsail bucket with multipart upload

With multipart upload, you can upload a single file to your bucket as a set of parts. Each part is a contiguous portion of the file's data. You can upload these file parts independently and in any order. If transmission of any part fails, you can retransmit that part without affecting other parts. After all parts of your file are uploaded, Amazon S3 assembles these parts and creates the object in your bucket in Amazon Lightsail. In general, when your object size reaches 100 MB, you should consider using multipart uploads instead of uploading the object in a single operation. For more information about buckets, see [Object storage](#).

Using multipart upload provides the following advantages:

- Improved throughput - You can upload parts in parallel to improve throughput.
- Quick recovery from any network issues - Smaller part size minimizes the impact of restarting a failed upload due to a network error.
- Upload over time - You can upload file parts over time. After you initiate a multipart upload, you have 24 hours to complete the multipart upload.
- Begin an upload before you know the final file size - You can upload a file as you are creating it.

We recommend that you use multipart upload in the following ways:

- If you're uploading large files over a stable high-bandwidth network, multipart upload maximizes the use of your available bandwidth by uploading file parts in parallel for multi-threaded performance.
- If you're uploading over a spotty network, use multipart upload to increase resiliency to network errors by avoiding upload restarts. When using multipart upload, you retry uploads only for the interrupted parts. There's no need to start over or upload the entire file again.

Contents

- [Multipart upload process](#)
- [Concurrent multipart upload operations](#)
- [Multipart upload retention](#)
- [Amazon Simple Storage Service multipart upload limits](#)
- [Split the file to upload](#)
- [Initiate a multipart upload using the AWS CLI](#)
- [Upload a part using the AWS CLI](#)
- [List parts of a multipart upload using the AWS CLI](#)
- [Create a multipart upload .json file](#)
- [Complete a multipart upload using the AWS CLI](#)
- [List multipart uploads for a bucket using the AWS CLI](#)
- [Stop a multipart upload using the AWS CLI](#)

Multipart upload process

Multipart upload is a three-step process that uses Amazon S3 actions to upload files to your bucket in Lightsail:

1. You initiate the multipart upload using the [CreateMultipartUpload](#) action.
2. You upload the file parts using the [UploadPart](#) action.
3. You complete the multipart upload using the [CompleteMultipartUpload](#) action.

Note

You can stop a multipart upload after you've initiated it by using the [AbortMultipartUpload](#) action.

When the multipart upload request completes, Amazon Simple Storage Service constructs the object from the uploaded parts. Then you can access the object in the same way that you would access any other object in your bucket.

You can list all of your in-progress multipart uploads or get a list of the parts that you have uploaded for a specific multipart upload. Each of these operations is explained in this section.

Multipart upload initiation

When you send a request to initiate a multipart upload, Amazon Simple Storage Service returns a response with an upload ID. This is a unique identifier for your multipart upload. You must include the upload ID whenever you upload parts, list the parts, complete an upload, or stop an upload. If you want to provide any metadata describing the object being uploaded, you must specify the metadata in the request to initiate multipart upload.

Parts upload

When uploading a part, in addition to the upload ID, you must specify a part number. You can choose any part number between 1 and 10,000. A part number uniquely identifies a part and its position in the object you are uploading. The part number that you choose doesn't need to be in a consecutive sequence (for example, it can be 1, 5, and 14). If you upload a new part using the same part number as a previously uploaded part, the previously uploaded part is overwritten.

Whenever you upload a part, Amazon Simple Storage Service returns an `#ETag#` header in its response. For each part upload, you must record the part number and the ETag value. You must include these values in the subsequent request to complete the multipart upload.

Note

All uploaded parts of a multipart upload are stored on your bucket. They consume your bucket's storage space until you complete the upload, stop the upload, or the upload times-out. For more information, see [Multipart upload retention](#) later in this guide.

Multipart upload completion

When you complete a multipart upload, Amazon Simple Storage Service creates an object by concatenating the parts in ascending order based on the part number. If any object metadata was provided in the `initiate multipart upload` request, Amazon Simple Storage Service associates that metadata with the object. After a successful `complete` request, the parts no longer exist.

Your `complete multipart upload` request must include the upload ID and a list of both part numbers and corresponding ETag values. The Amazon Simple Storage Service response includes an ETag that uniquely identifies the combined object data. This ETag is not necessarily an MD5 hash of the object data.

You can optionally stop the multipart upload. After stopping a multipart upload, you cannot upload any part using that upload ID again. All storage from any part of the canceled multipart upload is then freed. If any part uploads were in-progress, they can still succeed or fail even after you stop. To free all storage consumed by all parts, you must stop a multipart upload only after all part uploads have completed.

Multipart upload listings

You can list the parts of a specific multipart upload or all in-progress multipart uploads. The `list parts` operation returns the parts information that you have uploaded for a specific multipart upload. For each `list parts` request, Amazon Simple Storage Service returns the parts information for the specified multipart upload, up to a maximum of 1,000 parts. If there are more than 1,000 parts in the multipart upload, you must send a series of `list part` requests to retrieve all the parts. Note that the returned list of parts doesn't include parts that are still in the process of uploading. Using the `list multipart uploads` operation, you can obtain a list of multipart uploads in progress.

An in-progress multipart upload is an upload that you have initiated, but have not yet completed or stopped. Each request returns at most 1,000 multipart uploads. If there are more than 1,000 multipart uploads in progress, you must send additional requests to retrieve the remaining multipart uploads. Only use the returned listing for verification. Do not use the result of this listing when sending a `complete multipart upload` request. Instead, maintain your own list of the part numbers you specified when uploading parts and the corresponding ETag values that Amazon Simple Storage Service returns.

Concurrent multipart upload operations

In a distributed development environment, it is possible for your application to initiate several updates on the same object at the same time. Your application might initiate several multipart

uploads using the same object key. For each of these uploads, your application can then upload parts and send a complete upload request to Amazon Simple Storage Service to create the object. When the buckets have versioning enabled, completing a multipart upload always creates a new version. For buckets that don't have versioning enabled, other request might take precedence, such as requests that are received after a multipart upload is initiated and before it's complete.

Note

It is possible for other requests to take precedence, such as requests that are received after you initiate a multipart upload and before it is complete. For example, another operation might delete a key after you initiate a multipart upload with that key, and before the multipart upload is complete. If this occurs, the complete multipart upload response might indicate a successful object creation without you ever seeing the object.

Multipart upload retention

All uploaded parts of a multipart upload are stored on your bucket. They consume your bucket's storage space until you complete the upload, stop the upload, or the upload times out. A multipart upload times out, and the multipart upload is deleted, after 24 hours from when it was created. When you stop a multipart upload, or it times out, all uploaded parts are deleted and the storage space they used to consume on your bucket is freed.

Amazon Simple Storage Service multipart upload limits

The following table provides multipart upload core specifications.

- Maximum object size: 5 TB
- Maximum number of parts per upload: 10,000
- Part numbers: 1-10,000 (inclusive)
- Part size: 5 MB (minimum) - 5 GB (maximum). There is no size limit on the last part of your multipart upload.
- Maximum number of parts returned for a list parts request: 1,000
- Maximum number of multipart uploads returned in a list multipart uploads request: 1,000

Split the file to upload

Use the `split` command on the Linux or Unix operating system to split a file into multiple parts that you then upload to your bucket. There are similar free-ware applications that you can use on the Windows operating system to split a file. After you split the file into multiple parts, continue to the [Initiate a multipart upload](#) section of this guide.

Initiate a multipart upload using the AWS CLI

Complete the following procedure to initiate a multipart upload using the AWS Command Line Interface (AWS CLI). You do this by using the `create-multipart-upload` command. For more information, see [create-multipart-upload](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to create a multipart upload for your bucket.

```
aws s3api create-multipart-upload --bucket BucketName --key ObjectKey --acl bucket-owner-full-control
```

In the command, replace the following example text with your own:

- *BucketName*#- The name of the bucket for which you want to create a multipart upload.
- *ObjectKey*#- The object key to use for the file that you will upload.

Example:

```
aws s3api create-multipart-upload --bucket DOC-EXAMPLE-BUCKET --key sailbot.mp4 --acl bucket-owner-full-control
```

You should see a result similar to the following example. The response includes an UploadID, which you must specify in subsequent commands to upload parts, and to complete the multipart upload for this object.

```
C:\>aws s3api create-multipart-upload --bucket DOC-EXAMPLE-BUCKET --key sailbot.mp4
{
  "AbortDate": "2021-05-20T00:00:00+00:00",
  "AbortRuleId": "ExpireMultiPart",
  "ServerSideEncryption": "AES256",
  "Bucket": "DOC-EXAMPLE-BUCKET",
  "Key": "sailbot.mp4",
  "UploadId": "R4QU.m0.exampleIHWiL0eNw7JtXX70otRhTLsXXCzF21CZdYlfj51fjtIMnpzVw2WPj.exampleBTmL_N_.42.D1HY0TsITFsX.t03X0UTTAH1CxY5VR8jWRGdkVkuG"
}
```

After you have the UploadID for your multipart upload, continue to the following [Upload a part using the AWS CLI](#) section of this guide and start uploading parts.

Upload a part using the AWS CLI

Complete the following procedure to upload a part of a multipart upload using the AWS Command Line Interface (AWS CLI). You do this by using the `upload-part#` command. For more information, see [upload-part#](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to upload a part to your bucket.

```
aws s3api upload-part --bucket BucketName --key ObjectKey --part-number Number --
body FilePart --upload-id "UploadID" --acl bucket-owner-full-control
```

In the command, replace the following example text with your own:

- ***BucketName***#- The name of the bucket for which you want to create a multipart upload.
- ***ObjectKey***#- The object key to use for the file that you will upload.

- **Number** - The part number of the part you are uploading. A part number uniquely identifies a part and its position in the object you are uploading. Make sure to incrementally increase the `--part-number` parameter with each part that you upload. To do so, number them in the order in which Amazon Simple Storage Service should assemble the object when you complete the multipart upload.
- **FilePart** - The part file to upload from your computer.
- **UploadID** - The upload ID of the multipart upload that you created earlier in this guide.

Example:

```
aws s3api upload-part --bucket DOC-EXAMPLE-BUCKET --
key sailbot.mp4 --part-number 1 --body sailbot.mp4.001 --upload-id
"R4QU.m0.exampleIHWiLOeNw7JtXX70otRhTLsXXCzF21CZdYlfj51fjtiMnpzVw2WPj.exampleBTmL_N_.42.D1
--acl bucket-owner-full-control
```

You should see a result similar to the following example. Repeat the `upload-part` command for each part you upload. The response for each of your upload part requests will include an ETag value for the part that you uploaded. Record the ETag values for each of the parts that you upload. You will need all of the ETag values to complete the multipart upload, which is covered later in this guide.

```
C:\>aws s3api upload-part --bucket DOC-EXAMPLE-BUCKET --key sailbot.mp4 --part-number 1 --body sailbot.mp4.001
--upload-id "R4QU.m0.exampleIHWiLOeNw7JtXX70otRhTLsXXCzF21CZdYlfj51fjtiMnpzVw2WPj.exampleBTmL_N_.42.D1HY0TsITFsX.t03X0UTTAHiCxY5VR8jwRGdkvKUG"
{
  "ServerSideEncryption": "AES256",
  "ETag": "\"4example7530246113e837a860a38bbb\""
}
```

List parts of a multipart upload using the AWS CLI

Complete the following procedure to list parts of a multipart upload using the AWS Command Line Interface (AWS CLI). You do this by using the `list-parts#command`. For more information, see [list-parts#](#) in the *AWS CLI Command Reference*.

Complete this procedure to get the ETag values for all of the uploaded parts in a multipart upload. You will need these values to complete the multipart upload later in this guide. However, if you recorded all of the ETag values from the response of your part uploads, then you can skip this procedure and continue to the [Create a multipart upload .json](#) file section of this guide.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to list the parts of a multipart upload on your bucket.

```
aws s3api list-parts --bucket BucketName --key ObjectKey --upload-id "UploadID"
```

In the command, replace the following example text with your own:

- *BucketName*#- The name of the bucket for which you want to list the parts of a multipart upload.
- *ObjectKey*#- The object key of the multipart upload.
- *UploadID* - The upload ID of the multipart upload that you created earlier in this guide.

Example:

```
aws s3api list-parts --bucket DOC-EXAMPLE-BUCKET --key sailbot.mp4 --upload-id  
"R4QU.m0.exampleiHWiL0eNw7JtXX70otRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.DL"
```

You should see a result similar to the following example. The response lists all of the part numbers and ETag values for the parts that you uploaded in the multipart upload. Copy these values to your clipboard, and continue to the [Create a multipart upload .json](#) section of this guide.

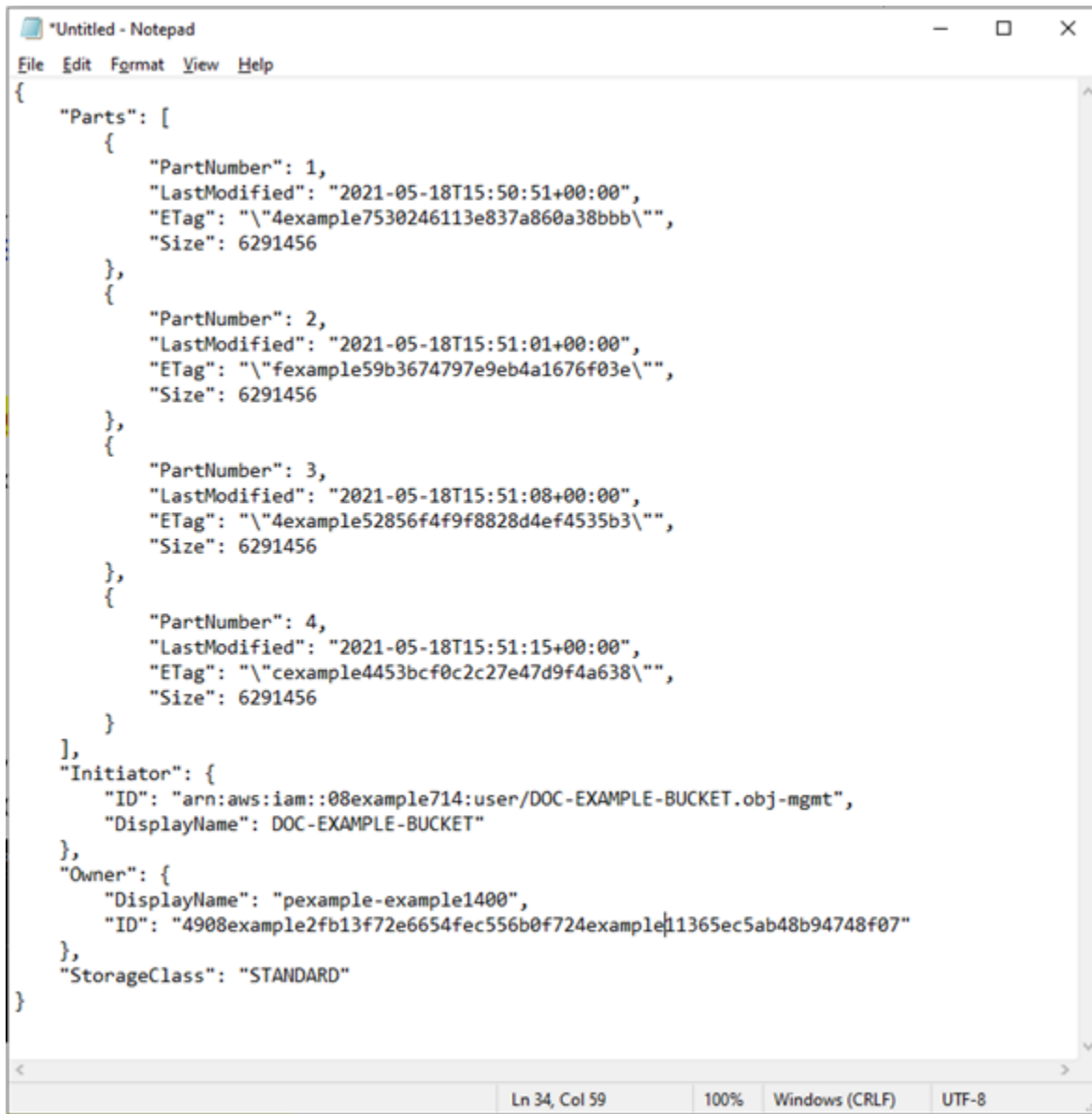
```
C:\>aws s3api list-parts --bucket DOC-EXAMPLE-BUCKET --key sailbot.mp4 --upload-id "R4QU.m0.exampleiHWiLOeNw7JtXX7OotR
hTLsXXCzF21CZdY1fj51fjtiMnpzVw2WPj.exampleBTmL_N_.42.D1HYOTsITFsX.t03XOUTTAHiCxY5VR8jWRGdkVkuG"
{
  "Parts": [
    {
      "PartNumber": 1,
      "LastModified": "2021-05-18T15:50:51+00:00",
      "ETag": "\"4example7530246113e837a860a38bbb\"",
      "Size": 6291456
    },
    {
      "PartNumber": 2,
      "LastModified": "2021-05-18T15:51:01+00:00",
      "ETag": "\"fexample59b3674797e9eb4a1676f03e\"",
      "Size": 6291456
    },
    {
      "PartNumber": 3,
      "LastModified": "2021-05-18T15:51:08+00:00",
      "ETag": "\"4example52856f4f9f8828d4ef4535b3\"",
      "Size": 6291456
    },
    {
      "PartNumber": 4,
      "LastModified": "2021-05-18T15:51:15+00:00",
      "ETag": "\"cexample4453bcf0c2c27e47d9f4a638\"",
      "Size": 6291456
    }
  ],
  "Initiator": {
    "ID": "arn:aws:iam:08example714:user/DOC-EXAMPLE-BUCKET.obj-mgmt",
    "DisplayName": "DOC-EXAMPLE-BUCKET"
  },
  "Owner": {
    "DisplayName": "pexample-example1400",
    "ID": "4908example2fb13f72e6654fec556b0f724example11365ec5ab48b94748f07"
  },
  "StorageClass": "STANDARD"
}
```

Create a multipart upload .json file

Complete the following procedure to create a multipart upload .json file that defines all of the parts you uploaded and their ETag values. This is required later in this guide to complete the multipart upload.

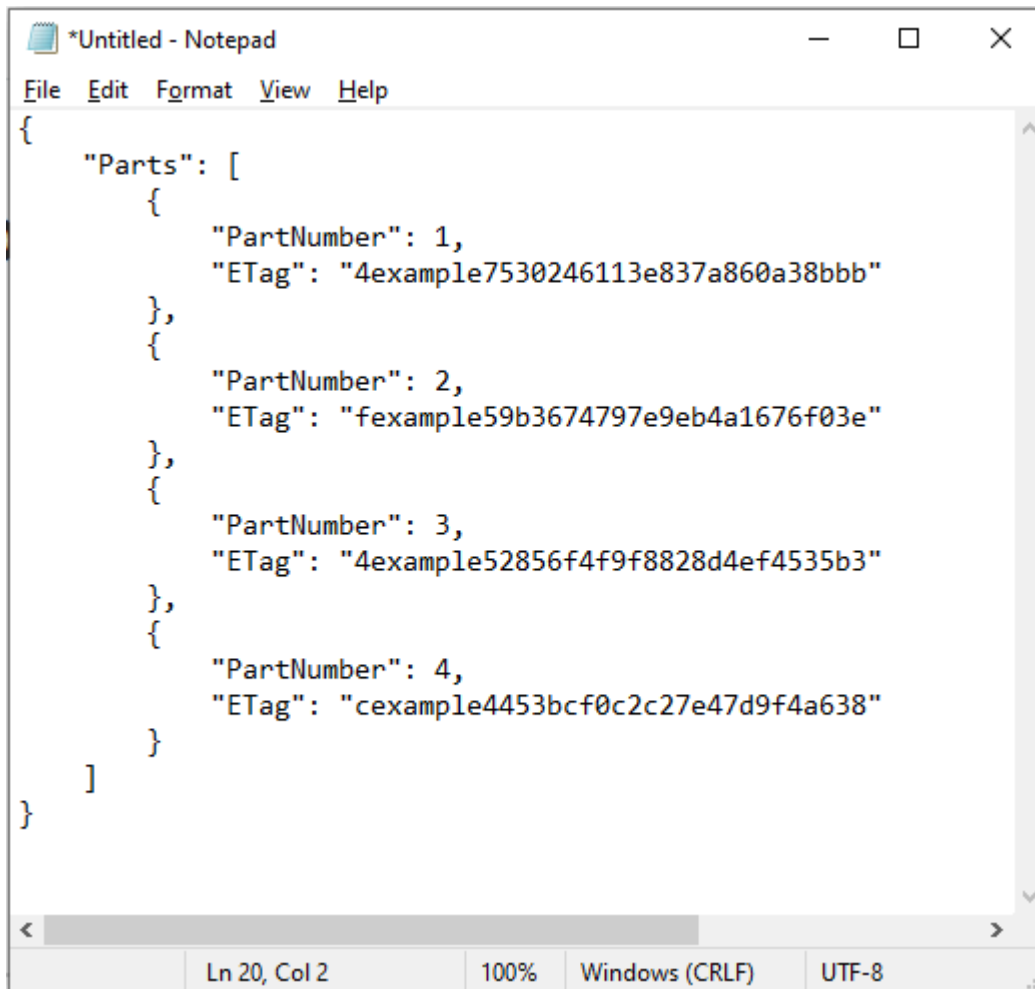
1. Open a text editor, and paste the response from the `list-parts` command that you requested in the previous section of this guide.

The result should look like the following example.

A screenshot of a Notepad window titled "Untitled - Notepad". The window contains a JSON object representing a multipart upload. The JSON is formatted with indentation and line breaks. The status bar at the bottom indicates "Ln 34, Col 59", "100%", "Windows (CRLF)", and "UTF-8".

```
{
  "Parts": [
    {
      "PartNumber": 1,
      "LastModified": "2021-05-18T15:50:51+00:00",
      "ETag": "\"4example7530246113e837a860a38bbb\"",
      "Size": 6291456
    },
    {
      "PartNumber": 2,
      "LastModified": "2021-05-18T15:51:01+00:00",
      "ETag": "\"fexample59b3674797e9eb4a1676f03e\"",
      "Size": 6291456
    },
    {
      "PartNumber": 3,
      "LastModified": "2021-05-18T15:51:08+00:00",
      "ETag": "\"4example52856f4f9f8828d4ef4535b3\"",
      "Size": 6291456
    },
    {
      "PartNumber": 4,
      "LastModified": "2021-05-18T15:51:15+00:00",
      "ETag": "\"cexample4453bcf0c2c27e47d9f4a638\"",
      "Size": 6291456
    }
  ],
  "Initiator": {
    "ID": "arn:aws:iam::08example714:user/DOC-EXAMPLE-BUCKET.obj-mgmt",
    "DisplayName": "DOC-EXAMPLE-BUCKET"
  },
  "Owner": {
    "DisplayName": "pexample-example1400",
    "ID": "4908example2fb13f72e6654fec556b0f724example11365ec5ab48b94748f07"
  },
  "StorageClass": "STANDARD"
}
```

2. Reformat the text file as shown in the following example:



```
*Untitled - Notepad
File Edit Format View Help
{
  "Parts": [
    {
      "PartNumber": 1,
      "ETag": "4example7530246113e837a860a38bbb"
    },
    {
      "PartNumber": 2,
      "ETag": "fexample59b3674797e9eb4a1676f03e"
    },
    {
      "PartNumber": 3,
      "ETag": "4example52856f4f9f8828d4ef4535b3"
    },
    {
      "PartNumber": 4,
      "ETag": "cexample4453bcf0c2c27e47d9f4a638"
    }
  ]
}
```

Ln 20, Col 2 100% Windows (CRLF) UTF-8

3. Save the text file to your computer as `mpstructure.json`, and continue to the [Complete a multipart upload using the AWS CLI](#) section of this guide.

Complete a multipart upload using the AWS CLI

Complete the following procedure to complete a multipart upload using the AWS Command Line Interface (AWS CLI). You do this by using the `complete-multipart-upload#command`. For more information, see [complete-multipart-upload#](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to upload a part to your bucket.

```
aws s3api complete-multipart-upload --multipart-upload file://JSONFileName --
bucket BucketName --key ObjectKey --upload-id "UploadID" --acl bucket-owner-full-
control
```

In the command, replace the following example text with your own:

- *JSONFileName*#- The name of the .json file that you created earlier in this guide (for example, mpstructure.json).
- *BucketName*#- The name of the bucket for which you want to complete a multipart upload.
- *ObjectKey*#- The object key of the multipart upload.
- *UploadID* - The upload ID of the multipart upload that you created earlier in this guide.

Example:

```
aws s3api complete-multipart-upload --multipart-upload file://mpstructure.json
--bucket DOC-EXAMPLE-BUCKET --key sailbot.mp4 --upload-id
"R4QU.m0.exampleiHWiLOeNw7JtXX70otRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.D1
--acl bucket-owner-full-control
```

You should see a response similar to the following example. This confirms that the multipart upload is completed. The object is now assembled and available in the bucket.

```
C:\>aws s3api complete-multipart-upload --multipart-upload file://mpstructure.json --bucket DOC-EXAMPLE-BUCKET --key sailbot.mp4
--upload-id "R4QU.m0.exampleiHWiLOeNw7JtXX70otRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.D1HYOTsITfsX.t03XOUTTAH1CxY5VR8jWRGdkVkuG"
{
  "ServerSideEncryption": "AES256",
  "VersionId": "MexampleKMdfPQb.2VZHqOvE_T.vSDtY",
  "Location": "https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/sailbot.mp4",
  "Bucket": "DOC-EXAMPLE-BUCKET",
  "Key": "sailbot.mp4",
  "ETag": "\"1example5964e3115e5d3f3c9a731585-4\""
}
```

List multipart uploads for a bucket using the AWS CLI

Complete the following procedure to list all multipart uploads for a bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `list-multipart-uploads#command`. For more information, see [list-multipart-uploads#](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to upload a part to your bucket.

```
aws s3api list-multipart-uploads --bucket BucketName
```

In the command, replace *BucketName* with the name of the bucket for which you want to list all multipart uploads.

Example:

```
aws s3api list-multipart-uploads --bucket DOC-EXAMPLE-BUCKET
```

You should see a response similar to the following example.

```
C:\>aws s3api list-multipart-uploads --bucket DOC-EXAMPLE-BUCKET
{
  "Uploads": [
    {
      "UploadId": "R4QU.m0.exampleiHwiL0eNw7JtXX70otRhTlsXXCzF21CzdYlfj51fjtiMnpzVw2WPj.exampleBTmL_N_.42.D1HYOTsITFsX.t03XOUTTAHicxY5VR8jWRGdkVkuG",
      "Key": "sailbot.mp4",
      "Initiated": "2021-05-18T15:49:11+00:00",
      "StorageClass": "STANDARD",
      "Owner": {
        "DisplayName": "pexample-example1400",
        "ID": "4908example2fb13f72e6654fec556b0f724example11365ec5ab48b94748f07"
      },
      "Initiator": {
        "ID": "arn:aws:iam::08example714:user/DOC-EXAMPLE-BUCKET.obj-mgmt",
        "DisplayName": "DOC-EXAMPLE-BUCKET"
      }
    }
  ]
}
```

Stop a multipart upload using the AWS CLI

Complete the following procedure to stop a multipart upload using the AWS Command Line Interface (AWS CLI). You do this if you started a multipart upload but no longer want to continue it. You do this by using the `abort-multipart-upload#command`. For more information, see [abort-multipart-upload#](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to upload a part to your bucket.

```
aws s3api abort-multipart-upload --bucket BucketName --key ObjectKey --upload-id
"UploadID" --acl bucket-owner-full-control
```

In the command, replace the following example text with your own:

- *BucketName*#- The name of the bucket for which you want to stop a multipart upload.
- *ObjectKey*#- The object key of the multipart upload.
- *UploadID* - The upload ID of the multipart upload that you want to stop.

Example:

```
aws s3api abort-multipart-upload --bucket DOC-EXAMPLE-BUCKET --key sailbot.mp4 --
upload-id
"R4QU.m0.exampleiHWiLOeNw7JtXX70otRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.DL"
--acl bucket-owner-full-control
```

This command does not return a response. You can run a `list-multipart-uploads#command` to confirm that the multipart upload was stopped.

Bucket naming rules in Amazon Lightsail

When you create a bucket in the Amazon Lightsail object storage service, you must give it a name. The name of the bucket is part of the URL that your customers will use when accessing objects that are stored in the bucket. For example, if you name your bucket `DOC-EXAMPLE-BUCKET` in the `us-east-1` AWS Region, the URL for your bucket is `DOC-EXAMPLE-BUCKET.s3.us-east-1.amazonaws.com`. You cannot change the name of your bucket after you create it. Keep in

mind that your customers are able to see the bucket name that you specify. For more information about the Lightsail object storage service, see [Object storage](#). For more information about creating buckets, see [Create a bucket](#).

Bucket names must be DNS-compliant. Because of this, the following rules apply for naming buckets in Lightsail:

- Bucket names must be between 3 and 56 characters long.
- Bucket names can consist only of lowercase letters, numbers, and hyphens (-).
- Bucket names must begin and end with a letter or number.
- Hyphens (-) can separate words, but cannot be specified consecutively. For example, doc-example-bucket is allowed but doc--example--bucket is not allowed.
- Bucket names must be unique within the aws (Standard Regions) partition, including buckets in Amazon Simple Storage Service (Amazon S3).

Example bucket names

The following example bucket names are valid and follow the recommended naming guidelines:

- docexamplebucket1
- log-delivery-march-2020
- my-hosted-content

The following example bucket names are not allowed:

- doc.example.bucket
- doc--example--bucket
- doc-example-bucket-

Key names for Lightsail object storage buckets

Files that you upload to your bucket are stored as objects in the Amazon Lightsail object storage service. An object key (or key name) uniquely identifies an object stored in a bucket. This guide explains the concept of key names and key name prefixes that make up the folder structure of

buckets viewed through the Lightsail console. For more information about buckets, see [Object storage](#).

Key names

The Lightsail object storage service data model uses a flat structure instead of a hierarchical structure like you would see in a file system. There is no hierarchy of folders and subfolders. However, you can infer logical hierarchy using key name prefixes and delimiters. The Lightsail console uses the key name prefixes to display your objects in a folder structure.

Suppose that your bucket has four objects with the following object keys:

- `Development/Projects.xls`
- `Finance/statement1.pdf`
- `Private/taxdocument.pdf`
- `to-dos.doc`

The Lightsail console uses the key name prefixes (`Development/`, `Finance/`, and `Private/`) and the delimiter (`/`) to present a folder structure. The `to-dos.doc` key name does not have a prefix, so its object appears directly at the root level of your bucket. If you browse to the `Development/` folder in the Lightsail console, you see the `Projects.xls` object. In the `Finance/` folder, you see the `statement1.pdf` object, and in the `Private/` folder, you see the `taxdocument.pdf` object.

The Lightsail console allows for folder creation by creating a zero-byte object with the key name prefix and delimiter value as the key name. These folder objects don't appear in the console. However, they behave like any other objects. You can view and manipulate them using the Amazon S3 API, AWS Command Line Interface (AWS CLI), or AWS SDKs.

Object key naming guidelines

You can use any UTF-8 character in an object key name. However, using certain characters in key names can cause problems with some applications and protocols. The following guidelines help you maximize compliance with DNS, web-safe characters, XML parsers, and other APIs.

Safe characters

The following character sets are generally safe for use in key names.

- Alphanumeric characters

- 0-9
- a-z
- A-Z
- Special characters
 - Forward slash (/)
 - Exclamation point (!)
 - Hyphen (-)
 - Underscore (_)
 - Period (.)
 - Asterisk (*)
 - Single quote (')
 - Open parenthesis ((
 - Close parenthesis ())

The following are examples of valid object key names:

- `4my-organization`
- `my.great_photos-2014/jan/myvacation.jpg`
- `videos/2014/birthday/video1.wmv`

Important

If an object key name ends with a single period (.), or two periods (..), you can't download the object using the Lightsail console. To download an object with a key name ending with one or two periods, you must use the Amazon S3 API, AWS CLI, and AWS SDKs. For more information, see [Download bucket objects](#).

Characters that might require special handling

The following characters in a key name might require additional code handling and likely need to be URL encoded or referenced as HEX. Some of these are non-printable characters that your browser might not handle, which also requires special handling:

- Ampersand ("&")
- Dollar ("\$")
- ASCII character ranges 00–1F hex (0–31 decimal) and 7F (127 decimal)
- 'At' symbol ("@")
- Equals ("=")
- Semicolon (";")
- Colon (":")
- Plus ("+")
- Space – Significant sequences of spaces might be lost in some uses (especially multiple spaces)
- Comma (",")
- Question mark ("?")

Characters to avoid

Avoid the following characters in a key name because of significant special handling for consistency across all applications.

- Backslash ("\\")
- Left curly brace ("{"
- Non-printable ASCII characters (128–255 decimal characters)
- Caret ("^")
- Right curly brace ("}")
- Percent character ("%")
- Grave accent / back tick ("`")
- Right square bracket ("]")
- Quotation marks
- 'Greater than' symbol (">")
- Left square bracket ("["
- Tilde ("~")
- 'Less than' symbol ("<")
- 'Pound' character ("#")
- Vertical bar / pipe ("|")

XML related object key constraints

As specified by the [XML standard on end-of-line handling](#), all XML text is normalized so that single carriage returns (ASCII code 13) and carriage returns immediately followed by a line feed (ASCII code 10) are replaced by a single line feed character. To ensure the correct parsing of object keys in XML requests, carriage returns and [other special characters must be replaced with their equivalent XML entity code](#) when they are inserted within XML tags. The following is a list of such special characters and their equivalent entity codes:

- ' as '
- " as "
- & as &
- < as <
- > as >
- \r as  or 
- \n as
 or

The following example illustrates the use of an XML entity code as a substitution for a carriage return. This DeleteObjects request deletes an object with the key parameter /some/prefix/objectwith\rcarriagereturn (where the \r is the carriage return).

```
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Object>
    <Key>/some/prefix/objectwith&#13;carriagereturn</Key>
  </Object>
</Delete>
```

Security Best Practices for object storage in Lightsail

Amazon Lightsail object storage provides a number of security features to consider as you develop and implement your own security policies. The following best practices are general guidelines and don't represent a complete security solution. Because these best practices might not be appropriate or sufficient for your environment, treat them as helpful considerations rather than prescriptions.

Contents

- [Preventative security best practices](#)
 - [Implement least privilege access](#)
 - [Verify that your Lightsail buckets are not publicly accessible](#)
 - [Enable block public access in Amazon S3](#)
 - [Attach instances to buckets to grant full programmatic access](#)
 - [Use cross-account access to give other AWS accounts access to objects in your bucket](#)
 - [Encryption of data](#)
 - [Enable versioning](#)
- [Monitoring and auditing best practices](#)
 - [Enable access logging and perform periodic security and access audits](#)
 - [Identify, tag, and audit your buckets](#)
 - [Implement monitoring using AWS monitoring tools](#)
 - [Use AWS CloudTrail](#)
 - [Monitor AWS security advisories](#)

Preventative security best practices

The following best practices can help prevent security incidents with Lightsail buckets.

Implement least privilege access

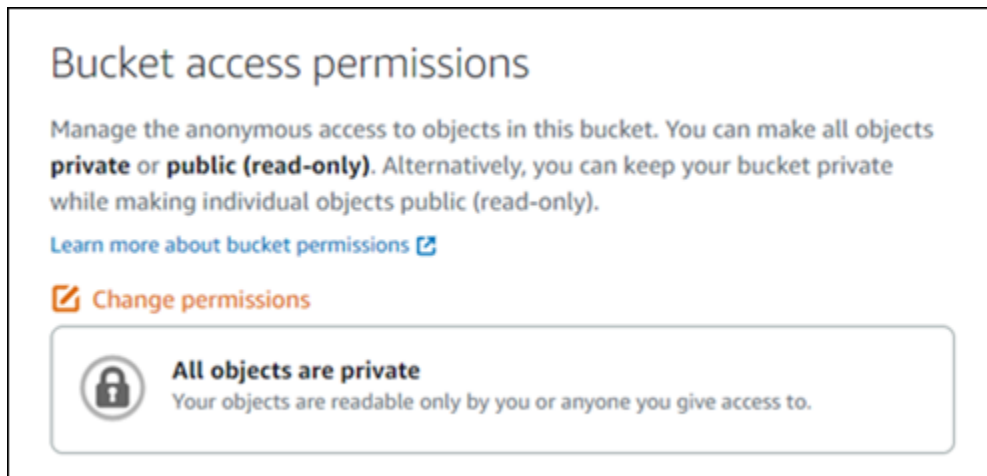
When granting permissions, you decide who is getting what permissions to which Lightsail resources. You enable specific actions that you want to allow on those resources. Therefore, you should grant only the permissions that are required to perform a task. Implementing least privilege access is fundamental in reducing security risk and the impact that could result from errors or malicious intent.

For more information about creating an IAM policy to manage buckets, see [IAM policy to manage buckets](#). For more information about the Amazon S3 actions supported by Lightsail buckets, see [Actions for object storage](#) in the *Amazon Lightsail API reference*.

Verify that your Lightsail buckets are not publicly accessible

Buckets and objects are private by default. Keep your bucket private by having the bucket access permission set to **All objects are private**. For the majority of use-cases, you don't need to make


your bucket or individual objects public. For more information, see [Configure access permissions for individual objects in a bucket](#).




Bucket access permissions

Manage the anonymous access to objects in this bucket. You can make all objects **private** or **public (read-only)**. Alternatively, you can keep your bucket private while making individual objects public (read-only).

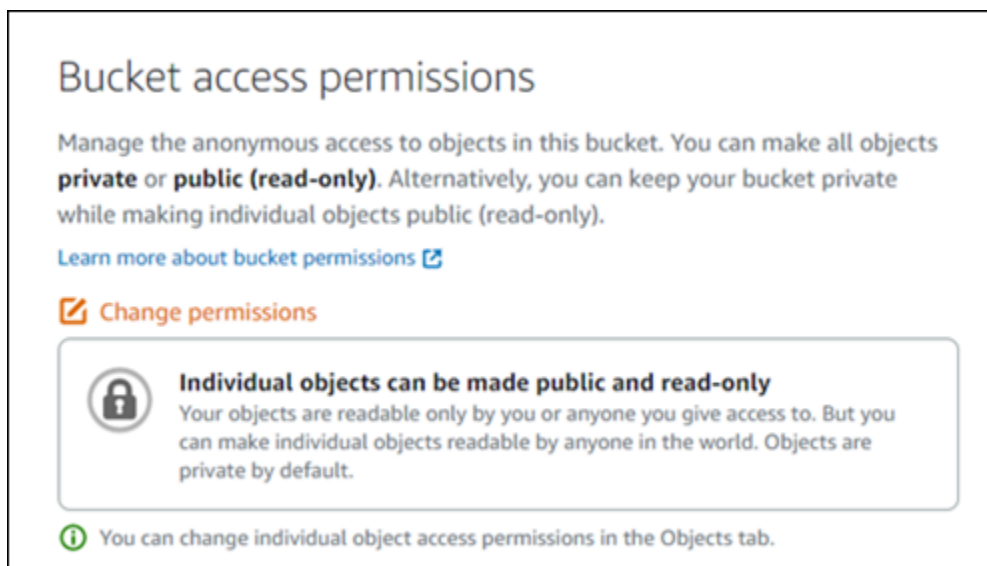
[Learn more about bucket permissions](#)

 **Change permissions**

 **All objects are private**
Your objects are readable only by you or anyone you give access to.

However, if you are using your bucket to host media for your website or application, under certain scenarios, you might need to make your bucket or individual objects public. You can configure one of the following options to make your bucket or individual objects public:


- If only some of the objects in a bucket need to be public (read-only) to anyone on the internet, then change the bucket access permission to **Individual objects can be made public and read-only**, and change only the objects that need to be public to **Public (read-only)**. This option keeps the bucket private, but gives you the option to make individual objects public. Don't make an individual object public if it contains sensitive or confidential information that you don't want to be publicly accessible. If you make individual objects public, you should periodically validate the public accessibility of each individual object.





Bucket access permissions

Manage the anonymous access to objects in this bucket. You can make all objects **private** or **public (read-only)**. Alternatively, you can keep your bucket private while making individual objects public (read-only).

[Learn more about bucket permissions](#)

 **Change permissions**

 **Individual objects can be made public and read-only**
Your objects are readable only by you or anyone you give access to. But you can make individual objects readable by anyone in the world. Objects are private by default.


 You can change individual object access permissions in the Objects tab.


- If all objects in the bucket need to be public (read-only) to anyone on the internet, then change the bucket access permission to **All objects are public and read-only**. Don't use this option if any of your objects in the bucket contain sensitive or confidential information.

Bucket access permissions

Manage the anonymous access to objects in this bucket. You can make all objects **private** or **public (read-only)**. Alternatively, you can keep your bucket private while making individual objects public (read-only).

[Learn more about bucket permissions](#)

 [Change permissions](#)


 **All objects are public and read-only**
Your objects are public (read-only) to anyone in the world.


- If you previously changed a bucket to be public, or changed individual objects to be public, you can quickly change the bucket and all its objects to be private by changing the bucket access permission to **All objects are private**.

Bucket access permissions

Manage the anonymous access to objects in this bucket. You can make all objects **private** or **public (read-only)**. Alternatively, you can keep your bucket private while making individual objects public (read-only).

[Learn more about bucket permissions](#)

 [Change permissions](#)

 **All objects are private**
Your objects are readable only by you or anyone you give access to.

Enable block public access in Amazon S3

Lightsail object storage resources take into account both Lightsail bucket access permissions and Amazon S3 account-level block public access configurations when allowing or denying public access. With Amazon S3 account-level block public access, account administrators and bucket owners can centrally limit public access to their Amazon S3 and Lightsail buckets. Block public access can make all Amazon S3 and Lightsail buckets private regardless of how the resources are

created, and regardless of the individual bucket and object permissions that might have been configured. For more information, see [Block public access for buckets](#).

Attach instances to buckets to grant full programmatic access

Attaching an instance to a Lightsail object storage bucket is the most secure way to provide access to the bucket. The **Resource access** functionality, which is how you attach an instance to a bucket, grants the instance full programmatic access to the bucket. With this method, you don't have to store bucket credentials directly in the instance or application, and you don't have to periodically rotate the credentials. For example, some WordPress plugins can access a bucket that the instance has access to. For more information, see [Configure resource access for a bucket](#) and [Tutorial: Connect a bucket to your WordPress instance](#).

Resource access

Attach instances to this bucket to give them access without the need to manage credentials.

[Learn more about resource access](#)

Attach instance

WordPress

1 GB RAM, 1 vCPU, 40 GB SSD

WordPress instance

Detach

However, if the application is not on a Lightsail instance, then you can create and configure bucket access keys. Bucket access keys are long term credentials that are not automatically rotated.

Access keys

Create access keys to generate credentials for this bucket that you can use in your code, plugins, and applications. You can have a maximum of 2 access keys at a time.

[Learn more about access keys](#)

Create access key

Access key ID	Secret access key	Created	Last used
AKIAIOSF0DNN7EXAMPLE	****	8/20/2021, 10:45 AM	—

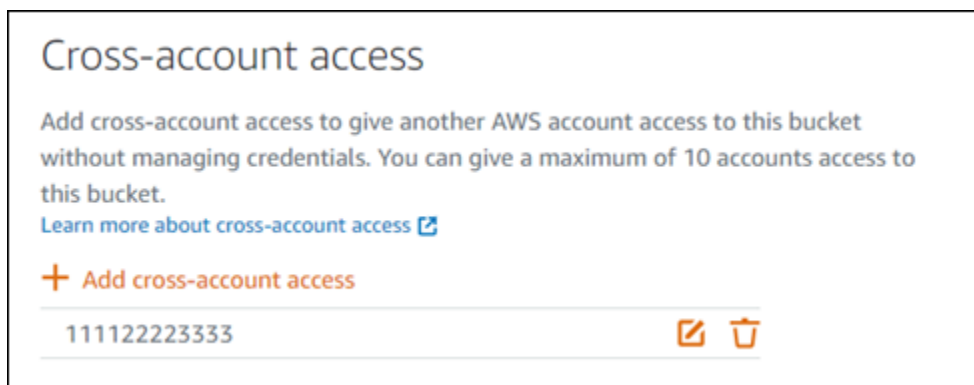
You can create and use access keys to grant applications or plugins full programmatic access to objects in your bucket. If you use an access key with your bucket, you should periodically rotate your keys and take inventory of the existing keys. Confirm the date an access key was last used, and

the AWS Region in which it was used, correspond with your expectations of how the key should be used. The date an access key was last used is displayed in the Lightsail console; in the **Access keys** section of the **Permissions** tab of a bucket's management page. Delete access keys that are not being used.

If you accidentally share your secret access key with the public, you should delete it and create a new one. You can have a maximum of two access keys per bucket. Even though you can have two different access keys at the same time, having one access key unused in your bucket is helpful for when you need to rotate a key with minimal downtime. To rotate an access key, create a new one, configure it on your software and test it, and then delete the earlier key. After you delete an access key, it's gone forever and can't be restored. It can only be replaced with a new access key. For more information, see [Create bucket access keys](#).

Use cross-account access to give other AWS accounts access to objects in your bucket

You can use cross-account access to make objects in a bucket accessible to a specific individual who has an AWS account without making the bucket and its objects public. If you've configured cross account access, make sure that the account IDs listed are the correct accounts that you want to give access to objects in your bucket. For more information, see [Configure cross-account access for a bucket](#).



Encryption of data

Lightsail performs server-side encryption with Amazon managed keys and encryption of data in transit by enforcing HTTPS (TLS). Server-side encryption helps reduce risk to your data by encrypting the data with a key that is stored in a separate service. In addition, encryption of data in transit helps prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle or similar attacks.

Enable versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Lightsail bucket. With versioning, you can easily recover from both unintended user actions and application failures. For more information, see [Enable and suspend bucket object versioning](#).

Monitoring and auditing best practices

The following best practices can help detect potential security weaknesses and incidents for Lightsail buckets.

Enable access logging and perform periodic security and access audits

Access logging provides detailed records for the requests that are made to a bucket. This information can include the request type (GET, PUT), the resources that are specified in the request, and the time and date that the request was processed. Enable access logging for a bucket, and periodically perform a security and access audit to identify the entities that are accessing your bucket. By default, Lightsail doesn't collect access logs for your buckets. You must manually enable access logging. For more information, see [Bucket access logs](#) and [Enable bucket access logging](#).

Identify, tag, and audit your Lightsail buckets

Identification of your IT assets is a crucial aspect of governance and security. You need to have visibility of all your Lightsail buckets to assess their security posture and take action on potential areas of weakness.

Use tagging to identify security-sensitive or audit-sensitive resources, then use those tags when you need to search for these resources. For more information, see [Tags](#).

Implement monitoring using AWS monitoring tools

Monitoring is an important part of maintaining the reliability, security, availability, and performance of Lightsail buckets and other resources. You can monitor and create notification alarms for the **Bucket size** (`BucketSizeBytes`) and **Number of objects** (`NumberOfObjects`) bucket metrics in Lightsail. For example, you might want to be notified when the size of your bucket increases or decreases to a specific size, or when the number of objects in your bucket goes up to or down to a specific number. For more information, see [Create bucket metric alarms](#).

Use AWS CloudTrail

AWS CloudTrail provides a record of actions taken by a user, a role, or an AWS service in Lightsail. You can use information collected by CloudTrail to determine the request that was made to Lightsail, the IP address from which the request was made, who made the request, when it was made, and additional details. For example, you can identify CloudTrail entries for actions that impact data access, in particular `CreateBucketAccessKey`, `GetBucketAccessKeys`, `DeleteBucketAccessKey`, `SetResourceAccessForBucket`, and `UpdateBucket`. When you set up your AWS account, CloudTrail is enabled by default. You can view recent events in the CloudTrail console. To create an ongoing record of activity and events for your Lightsail buckets, you can create a trail in the CloudTrail console. For more information, see [Logging Data Events for Trails](#) in the *AWS CloudTrail User Guide*.

Monitor AWS security advisories

Actively monitor the primary email address registered to AWS account. AWS will contact you, using this email address, about emerging security issues that might affect you.

AWS operational issues with broad impact are posted on the [AWS Service Health Dashboard](#). Operational issues are also posted to individual accounts via the Personal Health Dashboard. For more information, see the [AWS Health Documentation](#).

Understanding bucket permissions in Amazon Lightsail

By default, all Amazon Lightsail object storage resources—buckets and objects—are private. This means that only the bucket owner, the Lightsail account that created it, can access the bucket and its objects. The bucket owner can optionally grant access to others. You can grant access to a bucket and its objects in the following ways:

- **Read-only access** – The following options control read-only access to a bucket and its objects through the bucket's URL (for example, `https://DOC-EXAMPLE-BUCKET.us-east-1.amazonaws.com/media/sailbot.jpg`).
- **Bucket access permissions** – Use bucket access permissions to grant access to all objects in a bucket for anyone on the internet. For more information, see [Bucket access permissions](#) later in this guide.
- **Individual object access permissions** – Use individual object access permissions to grant access to an individual object in a bucket for anyone on the internet. For more information, see [Individual object access permissions](#) later in this guide.

- **Cross-account access** – Use cross-account access to grant access to all objects in a bucket for other AWS accounts. For more information, see [Cross-account access](#) later in this guide.
- **Read and write access** – The following options control full read and write access to a bucket and its objects. Use these options with the AWS Command Line Interface (AWS CLI), AWS APIs, and AWS SDKs.
 - **Access keys** – Use access keys to grant access to applications or plugins. For more information, see [Access keys](#) later in this guide.
 - **Resource access** – Use resource access to grant access to a Lightsail instance. For more information, see [Resource access](#) later in this guide.
- **Amazon Simple Storage Service block public access** – Use the Amazon Simple Storage Service (Amazon S3) account-level block public access feature to centrally limit public access to buckets in Amazon S3 and in Lightsail. Block public access can make all Amazon S3 and Lightsail buckets private regardless of the individual bucket and object permissions that might have been configured. For more information, see [Amazon S3 block public access](#) later in this guide.

For more information about buckets, see [Object storage](#). For more information about security best practices, see [Security Best Practices for object storage](#).

Bucket access permissions

Use bucket access permissions to control public (unauthenticated) read-only access to objects in a bucket. You can choose one of the following options when configuring bucket access permissions:

- **All objects are private** – All objects in the bucket are readable only by you or anyone you give access to. This option does not allow for individual objects to be made public (read-only).
- **Individual objects can be made public (read-only)** – Objects in the bucket are readable only by you or anyone you give access to, unless you specify an individual object as public (read-only). This option allows for individual objects to be made public (read-only). For more information, see [Individual object access permissions](#) later in this guide.
- **All objects are public (read-only)** – All objects in the bucket are readable by anyone on the internet. All objects in the bucket become readable by anyone on the internet through the URL of the bucket (for example, `https://DOC-EXAMPLE-BUCKET.us-east-1.amazonaws.com/media/sailbot.jpg`) when you choose this option.

For more information about configuring bucket access permissions, see [Configure bucket access permissions](#).

Individual object access permissions

Use individual object access permissions to control public (unauthenticated) read-only access to individual objects in a bucket. Individual object access permissions can be configured only when the [Bucket access permissions](#) of a bucket allow for individual objects to be made public (read-only). You can choose one of the following options when configuring access permissions for an individual object:

- **Private** – The object is readable only by you or anyone you give access to.
- **Public (read-only)** – The object is readable by anyone on the internet. The individual object becomes readable by anyone on the internet through the URL of the bucket (for example, `https://DOC-EXAMPLE-BUCKET.us-east-1.amazonaws.com/media/sailbot.jpg`).

For more information about configuring individual object access permissions, see [Configure access permissions for individual objects in a bucket](#).

Cross-account access

Use cross-account access to grant authenticated read-only access to all objects in a bucket for other AWS accounts and their users. Cross-account access is ideal if you want to share objects with another AWS account. When you grant cross-account access to another AWS account, users in that account have read-only access to objects in a bucket through the URL of the bucket (for example, `https://DOC-EXAMPLE-BUCKET.us-east-1.amazonaws.com/media/sailbot.jpg`). You can give access to a maximum of 10 AWS accounts.

For more information about configuring cross-account access, see [Configure cross-account access for a bucket](#).

Access keys

Use access keys to create a set of credentials that grant full read and write access to a bucket and its objects. Access keys consist of an access key ID and a secret access key as a set. You can have a maximum of two access keys per bucket. You can configure access keys on your application so that it can access your bucket and its objects using the AWS APIs, and AWS SDKs. You can also configure access keys on the AWS CLI.

For more information about creating access keys, see [Create access keys for a bucket](#).

Resource access

Use resource access to grant full read and write access to a bucket and its objects for Lightsail instances. With resource access, you don't have to manage credentials like access keys. To grant access to an instance, attach the instance to a bucket in the same AWS Region. To deny access, detach the instance from the bucket. Resource access is ideal if you're configuring an application on your instance to programmatically upload and access files on your bucket. One such use-case is to configure a WordPress instance to store media files on a bucket. For more information, see [Tutorial: Connect a bucket to your WordPress instance](#) and [Tutorial: Use a bucket with a content delivery network distribution](#).

For more information about configuring resource access, see [Configure resource access for a bucket](#).

Amazon S3 block public access

Use the Amazon S3 block public access feature to centrally limit public access to buckets in Amazon S3 and in Lightsail. Block public access can make all Amazon S3 and Lightsail buckets private regardless of the individual bucket and object permissions that might have been configured. You can use the Amazon S3 console, AWS CLI, AWS SDKs, and REST API to configure block public access settings for all buckets in your account, including those in the Lightsail object storage service. For more information, see [Block public access for buckets](#).

Upload files to an Amazon Lightsail bucket

When you upload a file to your bucket in the Amazon Lightsail object storage service, it is stored as an object. Objects consist of the file data and metadata that describe the object. You can have any number of objects in a bucket.

You can upload any file type—images, backups, data, movies—into a bucket. The maximum file size that you can upload by using the Lightsail console is 2 GB. To upload a larger file, use the Lightsail API, AWS Command Line Interface (AWS CLI), or AWS SDKs.

Lightsail offers the following options depending on the size of the file you want to upload:

- **Upload an object up to 2 GB in size using the Lightsail Console** — With the Lightsail console, you can upload a single object up to 2 GB in size. For more information, see [Upload files to a bucket using the Lightsail console](#) later in this guide.

- **Upload an object up to 5 GB in size with a single operation using the AWS SDKs, REST API, or AWS CLI** — With a single PUT operation, you can upload a single object up to 5 GB in size. For more information, see [Upload files to a bucket using the AWS CLI](#) later in this guide.
- **Upload an object in parts using the AWS SDKs, REST API, or AWS CLI** — Using the multipart upload API, you can upload a single large object, of 5 MB to 5 TB in size. The multipart upload API is designed to improve the upload experience for larger objects. You can upload an object in parts. These object parts can be uploaded independently, in any order, and in parallel. For more information, see [Upload files to a bucket using multipart upload](#).

For more information about buckets, see [Object storage](#).

Object key names and versioning

When you upload a file using the Lightsail console, the file name is used as the object key name. An object key (or key name) uniquely identifies an object stored in a bucket. The folder that the file is uploaded into, if any, is used as the key name prefix. For example, if you upload a file named `sailbot.jpg` to a folder in your bucket named `images`, the full object key name and prefix will be `images/sailbot.jpg`. However, the object is displayed in the console as `sailbot.jpg` in the `images` folder. For more information about object key names, see [Key names for object storage buckets](#).

When you upload a directory using the Lightsail console, all of the files and subfolders in the directory are uploaded to the bucket. Lightsail then assigns an object key name that is a combination of each of the uploaded file names and the folder name. For example, if you upload a folder named `images` that contains two files, `sample1.jpg` and `sample2.jpg`, Lightsail uploads the files and then assigns the corresponding key names, `images/sample1.jpg` and `images/sample2.jpg`. The objects are displayed in the console as `sample1.jpg` and `sample2.jpg` in the `images` folder.

If you upload a file with a key name that already exists, and your bucket *does not have versioning enabled*, the new uploaded object replaces the previous object. However, if your bucket *has versioning enabled*, Lightsail creates a new version of the object instead of replacing the existing object. For more information, see [Enable and suspend bucket object versioning](#).

Upload files to a bucket using the Lightsail console

Complete the following procedure to upload files and directories using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket that you want to upload files and folders into.
4. In the **Objects** tab, perform one of the following actions:
 - Drag and drop files and folders to the **Objects** page.
 - Choose **Upload**, and choose **File** to upload an individual file, or **Directory** to upload a folder and all of its contents.

Note

You can also create a folder in by choosing **Create new folder**. You can then browse into the new folder and upload files to it.

An **Upload successful** message is displayed when the upload completes.

Upload files to a bucket using the AWS CLI

Complete the following procedure to upload files and folders to a bucket using the AWS Command Line Interface (AWS CLI). You do this by using the `put-object` command. For more information, see [put-object](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Amazon S3 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to upload a file to your bucket.

```
aws s3api put-object --bucket BucketName --key ObjectKey --body LocalDirectory --acl bucket-owner-full-control
```

In the command, replace the following example text with your own:

- *BucketName* with the name of the bucket to which you want to upload the file.
- *ObjectKey* with the full object key of the object in your bucket.
- *LocalDirectoryFire* with the local directory folder path on your computer of the file to upload.

Example:

- On a Linux or Unix computer:

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg --body home/user/Pictures/sailbot.jpg --acl bucket-owner-full-control
```

- On a Windows computer:

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg --body "C:\Users\user\Pictures\sailbot.jpg" --acl bucket-owner-full-control
```

You should see a result similar to the following example:

```
C:\>aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg --body "C:\Users\user\Pictures\sailbot.jpg"
{
  "ETag": "\"694d34edexamp1ed92d64f342aa234c3\""
}
```

Configure the AWS CLI for IPv6-only requests

Amazon S3 supports bucket access over IPv6. You make requests with Amazon S3 API calls over IPv6 by using dual-stack endpoints. This section provides examples of how to make requests to a dual-stack endpoint, over IPv6. For more information, see [Using Amazon S3 dual-stack endpoints](#) in the *Amazon S3 User Guide*. For instructions on setting up the AWS CLI, see [Configuring the AWS Command Line Interface to work with Amazon Lightsail](#).

Important

The client and the network accessing the bucket must be enabled to use IPv6. For more information, see [IPv6 reachability](#).

There are two ways to make S3 requests from an IPv6-only instance. You can configure the AWS CLI to direct all Amazon S3 requests to the dual-stack endpoint for the specified AWS Region. Or, if you want to use a dual-stack endpoint for specified AWS CLI commands only (not all commands), you can add the S3 dual-stack endpoint to every command.

Configure the AWS CLI

Set the configuration value `use_dualstack_endpoint` to `true` in a profile in your AWS Config file to direct all Amazon S3 requests made by the Amazon S3 and `s3api` AWS CLI commands to the dual-stack endpoint for the specified Region. You specify the Region in the AWS CLI config file, or in a command using the `--region` option.

Enter the following commands to configure the AWS CLI.

```
aws configure set default.s3.use_dualstack_endpoint true
```

```
aws configure set default.s3.addressing_style virtual
```

Add the dual-stack endpoint to a specific command

You can use the dual-stack endpoint per command by setting the `--endpoint-url` parameter to `https://s3.dualstack.aws-region.amazonaws.com` or `http://s3.dualstack.aws-region.amazonaws.com` for any `s3` or `s3api` command. In the example below, replace *bucketname* and *aws-region* with the name of your bucket and your AWS Region.

```
aws s3api list-objects --bucket bucketname --endpoint-url https://s3.dualstack.aws-region.amazonaws.com
```

Managing buckets and objects in Lightsail

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).

3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
 6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
 7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
 8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)

- [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
- 10 After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
- 11 Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
- 12 Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
- 13 Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
- 14 Learn how to connect your bucket to other resources. For more information, see the following tutorials.
- [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
- 15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Container services in Amazon Lightsail

An Amazon Lightsail container service is a highly scalable compute and networking resource on which you can deploy, run, and manage containers. A container is a standard unit of software that packages code and its dependencies together so the application runs quickly and reliably from one computing environment to another.

You can think of your Lightsail container service as a computing environment that lets you run containers on AWS infrastructure by using images that you create on your local machine and push to your service, or images from an online repository, like Amazon ECR Public Gallery.

You can also run containers locally, on your local machine, by installing software such as Docker. Amazon Elastic Container Service (Amazon ECS) and Amazon Elastic Compute Cloud (Amazon EC2) are other resources within the AWS infrastructure on which you can run containers. For more information, see the [Amazon ECS Developer Guide](#).

Contents

- [Containers](#)
- [Lightsail container service elements](#)
 - [Lightsail container services](#)
 - [Container service capacity \(scale and power\)](#)
 - [Pricing](#)
 - [Deployments](#)
 - [Deployment versions](#)
 - [Container image sources](#)
 - [Public endpoints and default domains](#)
 - [Custom domains and SSL/TLS certificates](#)
 - [Container logs](#)
 - [Metrics](#)
- [Use Lightsail container services](#)

Containers

A container is a standard unit of software that packages code and its dependencies together so the application runs quickly and reliably from one computing environment to another. You could run a container on your development environment, deploy it to your pre-production environment, and then deploy it to your production environment. Your containers will run reliably regardless of whether your development environment is your local machine, your pre-production environment is a physical server in a data center, or your production environment is a virtual private server in the cloud.

A container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime. By containerizing the application and its dependencies, you no longer have to worry about whether your software runs correctly on the operating system and infrastructure that you deploy it on – you can spend more time focusing on the code.

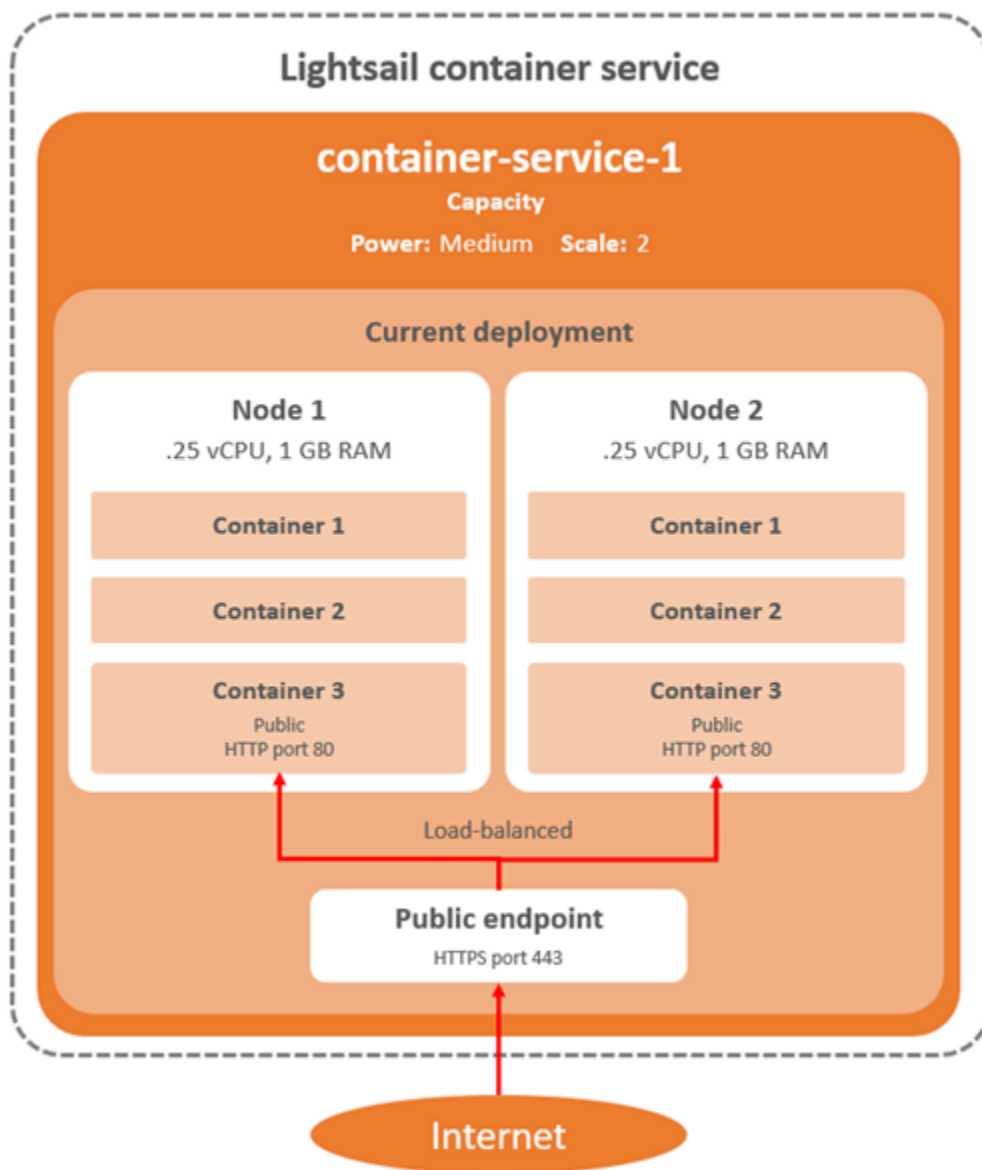
For more information about containers, and container images, see [What is a Container?](#) in the *Docker documentation*.

Lightsail container service elements

The following are the key elements of Lightsail container services that you should understand before getting started.

Lightsail container services

A container service is the Lightsail compute resource that you can create in any AWS Region in which Lightsail is available. You can create and delete container services at any time. For more information, see [Create Lightsail container services](#) and [Delete Lightsail container services](#).



Container service capacity (scale and power)

You must choose the following capacity parameters when you first create your container service:

- **Scale** — The number of compute nodes that you want your container workload to run in. Your container workload is copied across the compute nodes of your service. You can specify up to 20 compute nodes for a container service. You pick the scale based on the number of nodes you want powering your service for better availability and higher capacity. Traffic to your containers will be load-balanced across all nodes.

- **Power** — The memory and vCPUs of each node in your container service. The powers that you can choose are Nano (Na), Micro (Mi), Small (Sm), Medium (Md), Large (Lg), and Xlarge (Xl), each with a progressively greater amount of memory and vCPUs.

If you specify the scale of your container service as more than 1, then your container workload is copied across the multiple compute nodes of your service. For example, if the scale of your service is 3 and the power is Nano, then there are three copies of your container workload running on three compute resources each with 512 MB of RAM and 0.25 vCPUs. The incoming traffic is load-balanced between the three resources. The greater the capacity you specify for your container service, the more traffic it is able to handle.

You can dynamically increase the power and scale of your container service at any time without any down-time if you find that it's under-provisioned, or decrease it if you find that it's over-provisioned. Lightsail automatically manages the capacity change along with your current deployment. For more information, see [Change the capacity of your container service](#).

Pricing

The monthly price of your container service is calculated by multiplying the price of its power with the number of its compute nodes (the scale of your service). For example, a service with a medium power, which has a price of \$40 USD, and a scale of 3 compute nodes, will cost \$120 USD per month. You are charged for your container service whether it's enabled or disabled, and whether it has a deployment or not. You must delete your container service to stop being charged for it.

Each container service, regardless of its configured capacity, includes a monthly data transfer quota of 500 GB. The data transfer quota does not change regardless of the power and scale that you choose for your service. Data transfer out to the internet in excess of the quota will result in an overage charge that varies by AWS Region and starts at \$0.09 USD per GB. Data transfer in from the internet in excess of the quota does not incur an overage charge. For more information, see the [Lightsail pricing page](#).


Deployments

You can create a deployment in your Lightsail container service. A deployment is a set of specifications for the container workload that you wish to launch on your service.

You can specify the following parameters for each container entry in a deployment:

- The name of your container that will be launched

- The source container image to use for your container
- The command to run when launching your container
- The environment variables to apply to your container
- The network ports to open on your container
- The container in the deployment to make publicly accessible through the default domain of the container service

 **Note**

Only one container in a deployment can be made publicly accessible for each container service.

The following health check parameters will apply to the public endpoint of a deployment after it's launched:

- The directory path on which to perform a health check.
- Advanced health check settings, such as interval seconds, timeout seconds, success codes, healthy threshold, and unhealthy threshold.

Your container service can have one active deployment at a time, and a deployment can have up to 10 container entries. You can create a deployment at the same time as you create your container service, or you can create it after your service is up and running. For more information, see [Create and manage container service deployments](#).

Deployment versions

Every deployment that you create in your container service is saved as a deployment version. If you modify the parameters of an existing deployment, the containers are re-deployed to your service and the modified deployment results in a new deployment version. The latest 50 deployment versions for each container service are saved. You can use any of the 50 deployment versions to create a new deployment in the same container service. For more information, see [Create and manage container service deployments](#).

Container image sources

When you create a deployment, you must specify a source container image for each container entry in your deployment. Immediately after you create your deployment, your container service pulls the images from the sources you specify and uses them to create your containers.

The images that you specify can originate from the following sources:

- **A public registry**, such as Amazon ECR Public Gallery, or some other public container image registry. For more information about Amazon ECR Public, see [What Is Amazon Elastic Container Registry Public?](#) in the *Amazon ECR Public User Guide*.
- **Images pushed from your local machine** to your container service. If you create container images on your local machine, you can push them to your container service to use them when creating a deployment. For more information, see [Create container service images](#) and [Push and manage container images](#).

Lightsail container services support Linux-based container images. Windows-based container images are currently not supported, but you can run Docker, the AWS Command Line Interface (AWS CLI), and the Lightsail Control (lightsailctl) plugin on Windows to build and push your Linux based images to your Lightsail container service.

Public endpoints and default domains

When you create a deployment, you can specify the container entry in the deployment that will serve as the public endpoint of your container service. The application on the public endpoint container is publicly accessible on the internet through a randomly generated default domain of your container service. The default domain is formatted as `https://<ServiceName>.<RandomGUID>.<AWSRegion>.cs.amazonlightsail.com`, in which `<ServiceName>` is the name of your container service, `<RandomGUID>` is a randomly generated globally unique identifier of your container service in the AWS Region for your Lightsail account, and `<AWSRegion>` is the AWS Region in which the container service was created. The public endpoint of Lightsail container services supports HTTPS only, and it does not support TCP or UDP traffic. Only one container can be the public endpoint for a service. So make sure that choose the container that is hosting the front-end of your application as the public endpoint while rest of the containers are internally accessible.

You can use the default domain of your container service, or you can use your own custom domain (your registered domain name). For more information about using custom domains with your container services, see [Enable and manage custom domains for your container services](#).

Private domain

All container services also have a private domain that is formatted as `<ServiceName>.service.local`, in which `<ServiceName>` is the name of your container service. Use the private domain to access your container service from another one of your Lightsail resources in the same AWS Region as your service. The private domain is the only way to access your container service if you don't specify a public endpoint in the deployment of your service. A default domain is generated for your container service even if you don't specify a public endpoint, but it will show a 404 No Such Service error message when you try to browse to it.

To access a specific container using the private domain of your container service, you must specify the open port of the container that will accept your connection request. You do this by formatting the domain of your request as `<ServiceName>.service.local:<PortNumber>`, in which `<ServiceName>` is the name of your container service and `<PortNumber>` is the open port of the container that you wish to connect to. For example, if you create a deployment on your container service named `container-service-1`, and you specify a Redis container with port 6379 open, then you should format the domain of your request as `container-service-1.service.local:6379`.

Custom domains and SSL/TLS certificates

You can use up to 4 of your custom domains with your container service instead of using the default domain. For example, you can direct traffic for your custom domain, such as `example.com`, to the container in your deployment that is labeled as the public endpoint.

To use your custom domains with your service, you must first request an SSL/TLS certificate for the domains that you want to use. You must then validate the SSL/TLS certificate by adding a set of CNAME records to the DNS of your domains. After the SSL/TLS certificate is validated, you enable custom domains on your container service by attaching the valid SSL/TLS certificate to your service. For more information see [Create SSL/TLS certificates for your Lightsail container services](#), [Validate SSL/TLS certificates for your Lightsail container services](#), and [Enable and manage custom domains for your Lightsail container services](#).

Container logs

Every container in your container service generates a log that you can access to diagnose the operation of your containers. The logs provide the *stdout* and *stderr* streams of processes that run inside the container. For more information, see [View container service logs](#).

Metrics

Monitor the metrics of your container service to diagnose issues that may be a result of over-utilization. You can also monitor metrics to help you determine if your service is under-provisioned or over-provisioned. For more information, see [View container service metrics](#).

Use Lightsail container services

These are the general steps to manage your Lightsail container service if you plan to push container images from your local machine to your service, and use them in your deployment:

1. Create your container service in your Lightsail account. For more information, see [Create Lightsail container services](#).
2. Install software on your local machine that you need to create your own container images and push them to your Lightsail container service. For more information, see For more information, see the following guides:
 - [Install software to manage container images for your Lightsail container services](#)
 - [Create container images for your Lightsail container services](#)
 - [Push and manage container images on your Lightsail container services](#)
3. Create a deployment in your container service that configures and launches your containers. For more information, see [Create and manage deployments for your Lightsail container services](#).
4. View previous deployments for your container service. You can create a new deployment using a previous deployment version. For more information, see [View and manage deployment versions of your Lightsail container services](#).
5. View the logs of containers on your container service. For more information, see [View the container logs of your Lightsail container services](#).
6. Create an SSL/TLS certificate for the domains that you want to use with your containers. For more information, see [Create SSL/TLS certificates for your Lightsail container services](#).
7. Validate the SSL/TLS certificate by adding records to the DNS of your domains. For more information, see [Validate SSL/TLS certificates for your Lightsail container services](#).

8. Enable custom domains by attaching a valid SSL/TLS certificate to your container service. For more information, see [Enable and manage custom domains for your Lightsail container services](#).
9. Monitor the utilization metrics of your container service. For more information, see [View container service metrics](#).
- 10(Optional) Scale the capacity of your container service vertically, by increasing its power specification, and horizontally, by increasing its scale specification. For more information, see [Change the capacity of your Lightsail container services](#).
- 11Delete your container service if you're not using it to avoid incurring monthly charges. For more information, see [Delete Lightsail container services](#).

These are the general steps to manage your Lightsail container service if you plan to use container images from a public registry in your deployment:

1. Create your container service in your Lightsail account. For more information, see [Create Lightsail container services](#).
2. If you plan to use container images from a public registry, find container images from a public registry such as the Amazon ECR Public Gallery. For more information about Amazon ECR Public, see [What Is Amazon Elastic Container Registry Public?](#) in the *Amazon ECR Public User Guide*.
3. Create a deployment in your container service that configures and launches your containers. For more information, see [Create and manage deployments for your Lightsail container services](#).
4. View previous deployments for your container service. You can create a new deployment using a previous deployment version. For more information, see [View and manage deployment versions of your Lightsail container services](#).
5. View the logs of containers on your container service. For more information, see [View the container logs of your Lightsail container services](#).
6. Create an SSL/TLS certificate for the domains that you want to use with your containers. For more information, see [Create SSL/TLS certificates for your Lightsail container services](#).
7. Validate the SSL/TLS certificate by adding records to the DNS of your domains. For more information, see [Validate SSL/TLS certificates for your Lightsail container services](#).
8. Enable custom domains by attaching a valid SSL/TLS certificate to your container service. For more information, see [Enable and manage custom domains for your Lightsail container services](#).
9. Monitor the utilization metrics of your container service. For more information, see [View container service metrics](#).

10(Optional) Scale the capacity of your container service vertically, by increasing its power specification, and horizontally, by increasing its scale specification. For more information, see [Change the capacity of your Lightsail container services](#).

11Delete your container service if you're not using it to avoid incurring monthly charges. For more information, see [Delete Lightsail container services](#).

Create a Lightsail container service

In this guide, we show you how to create an Amazon Lightsail container service using the Lightsail console, and describe the container service settings that you can configure.

Before getting started, we recommend that you familiarize yourself with the elements of a Lightsail container service. For more information, see [Container services](#).

Container service capacity (scale and power)

You must choose the capacity of your container service when you first create it. The capacity is made up of a combination of the following parameters:

- **Scale** - The number of compute nodes that you want your container workload to run in. Your container workload is copied across the compute nodes of your service. You can specify up to 20 compute nodes for a container service. You pick the scale based on the number of nodes you want powering your service for better availability and higher capacity. Traffic to your containers will be load-balanced across all nodes.
- **Power** - The memory and vCPUs of each node in your container service. The powers that you can choose are Nano (Na), Micro (Mi), Small (Sm), Medium (Md), Large (Lg), and Xlarge (Xl); each with a progressively greater amount of memory and vCPUs.

The incoming traffic is load balanced across the scale (the number of compute nodes) of your container service. For example, a service with a Nano power and a scale of 3 will have 3 copies of your container workload running. Each node will have 512 MB of RAM and 0.25 vCPUs. The incoming traffic will be load-balanced across the 3 nodes. The greater the capacity you choose for your container service, the more traffic it is able to handle.

You can dynamically increase the power and scale of your container service at any time without any down-time if you find that it's under-provisioned, or decrease it if you find that it's over-

provisioned. Lightsail automatically manages the capacity change along with your current deployment. For more information, see [Change the capacity of your Lightsail container services](#).

Pricing

The monthly price of your container service is calculated by multiplying the base price of its power with the scale (the number of compute nodes). For example, a service with the \$40 USD medium power and a scale of 3, will cost \$120 USD per month.

Each container service, regardless of its configured capacity, includes a monthly data transfer quota of 500 GB. The data transfer quota does not change regardless of the power and scale that you choose for your service. Data transfer out to the internet in excess of the quota will result in an overage charge that varies by AWS Region and starts at \$0.09 USD per GB. Data transfer in from the internet in excess of the quota does not incur an overage charge. For more information, see the [Lightsail pricing page](#).

You are charged for your container service whether it's enabled or disabled, and whether it has a deployment or not. You must delete your container service to stop being charged for it. For more information, see [Delete Lightsail container services](#).

Container service status

Your container service can be in one of the following states:

- **Pending** – Your container service is being created.
- **Ready** – Your container service is running but it does not have an active container deployment.
- **Deploying** – Your deployment is being launched to your container service.
- **Running** – Your container service is running and it has an active container deployment.
- **Updating** – Your container service capacity or its custom domains are being updated.
- **Deleting** – Your container service is being deleted. Your container service is in this state after you choose to delete, and it's in this state only for a brief moment.
- **Disabled** – Your container service is disabled, and its active deployment and containers, if any, are shut down.

Container service sub-status

If your container service is in a **Deploying** or **Updating** state, then one of the following additional sub-states is displayed below the container service state:

- **Creating system resources** - The system resources for your container service are being created.
- **Creating network infrastructure** - The network infrastructure for your container service are being created.
- **Provisioning certificate** - The SSL/TLS certificate for your container service is being created.
- **Provisioning service** - Your container service is being provisioned.
- **Creating deployment** - Your deployment is being created on your container service.
- **Evaluating health check** - The health of your deployment is being evaluated.
- **Activating deployment** - Your deployment is being activated.

If your container service is in a **Pending** state, then one of the following additional sub-states is displayed below the container service state:

- **Certificate limit exceeded** - The SSL/TLS certificate required for your container service exceeds the maximum number of certificates allowed for your account.
- **Unknown error** - An error was experienced when your container service was being created.

Create a container service

Complete the following procedure to create a Lightsail container service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose **Create container service**.
4. In the **Create a container service** page, choose **Change AWS Region**, then choose an AWS Region for your container service.
5. Choose a capacity for your container service. For more information, see the [Container service capacity \(scale and power\)](#) section of this guide.
6. Complete the following steps to create a deployment that will be launched at the same time as your container service is created. Otherwise, skip to step 7 to create a container service without a deployment.

Create a container service with a deployment if you plan to use a container image from a public registry. Otherwise, create your service without a deployment if you plan to use a container image that is on your local machine. You can push the container image from your local machine to your container service after your service is up and running. Then you can

create a deployment using the pushed container image that is registered to your container service.

- a. Choose **Create a deployment**.
- b. Choose one of the following options:
 - **Choose an example deployment** – Choose this option to create a deployment using a container image that's been curated by the Lightsail team with a set of preconfigured deployment parameters. This option provides the fastest and easiest way to get a popular container up and running on your container service.
 - **Specify a custom deployment** – Choose this option to create a deployment by specifying containers of your choosing.

The deployment form view opens, where you can enter new deployment parameters.

- c. Enter the parameters of your deployment. For more information about the deployment parameters that you can specify, see the **Deployment parameters** section in the [Create and manage deployments for your Lightsail container services](#) guide.
 - d. Choose **Add container entry** to add more than one container entry to your deployment. You can have up to 10 container entries in your deployment.
 - e. When you're done entering the parameters of your deployment, choose **Save and deploy** to create the deployment on your container service.
7. Enter a name for your container service.

Container service names must be:

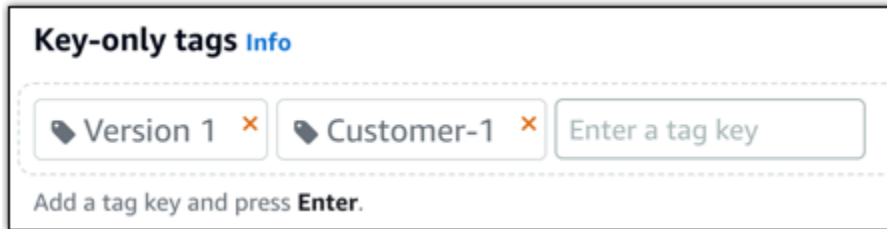
- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 63 characters.
- Must contain only alphanumeric characters and hyphens.
- A hyphen (-) can separate words but cannot be at the start or end of the name.

 **Note**

The name that you specify will be part of the default domain name of your container service, and it will be visible to the public.

8. Choose one of the following options to add tags to your container service:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



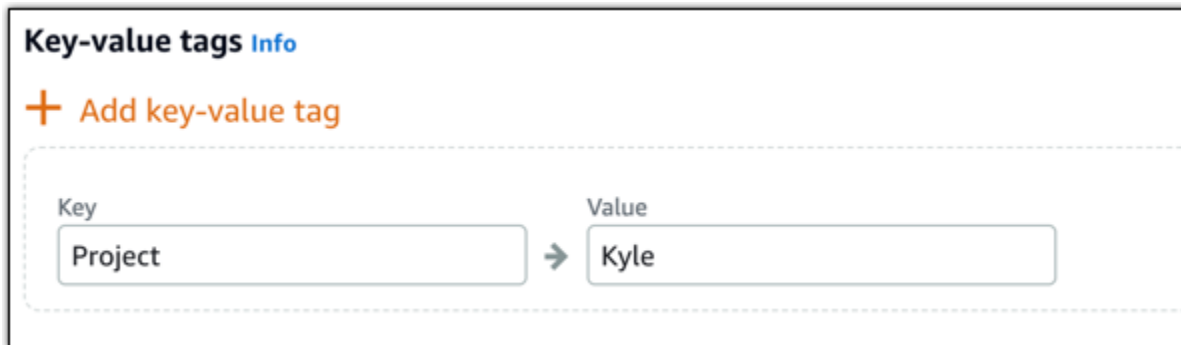
Key-only tags Info

Version 1 × Customer-1 × Enter a tag key

Add a tag key and press **Enter**.

- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key Value

Project → Kyle

Note

For more information about key-only and key-value tags, see [Tags](#).

9. Choose **Create container service**.

You are redirected to the management page of your new container service. The status of your new container service is **Pending** while it's being created. After a few moments, the status of your service changes to **Ready**, if it doesn't have a current deployment, or **Running**, if you created a deployment.

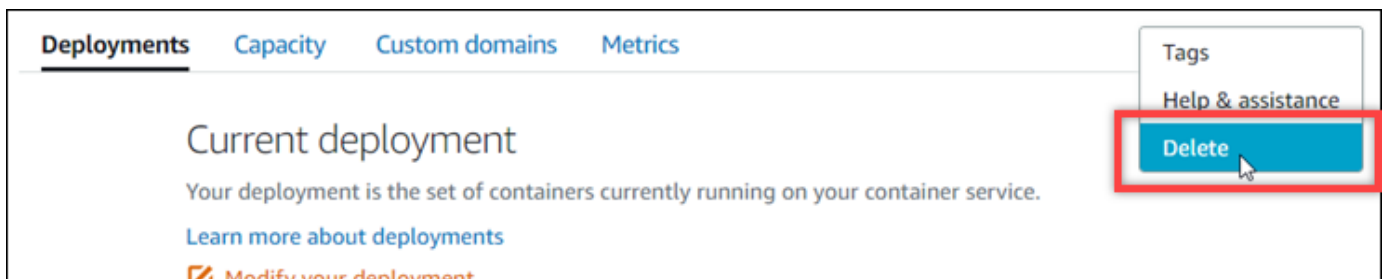
Delete a Lightsail container service

You can delete your Amazon Lightsail container service at any time if you're no longer using it. When you delete your container service, all deployments and registered container images associated with that service are permanently destroyed. However, the SSL/TLS certificates and domains that you created remain in your Lightsail account so that you can use them with another resource. For more information about container services, see [Container services in Amazon Lightsail](#).

Delete a container service

Complete the following procedure to delete your container service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service you want to delete.
4. Choose the ellipsis icon in the tab menu, then choose the **Delete**.



5. Choose **Delete container service** to delete your service.
6. In the prompt that appears, choose **Yes, delete** to confirm that the deletion is permanent.

Your container service is deleted after a few moments.

Lightsail container service images

With Docker, you can build, run, test, and deploy distributed applications that are based on containers. Amazon Lightsail container services use Docker container images in deployments to launch containers.

In this guide, we show you how to create a container image on your local machine using a Dockerfile. After your image is created, you can then push it to your Lightsail container service to deploy it.

To complete the procedures in this guide you should possess a basic understanding of what Docker is and how it works. For more information about Docker, see [What is Docker?](#) and the [Docker overview](#).

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Create a Dockerfile and build a container image](#)
- [Step 3: Run your new container image](#)
- [\(Optional\) Step 4: Clean up the containers running on your local machine](#)
- [Next steps after creating container images](#)

Step 1: Complete the prerequisites

Before you get started, you must install the software required to create containers and then push them to your Lightsail container service. For example, you must install and use Docker to create and build your container images that you can then use with your Lightsail container service. For more information, see [Installing software to manage container images for your Amazon Lightsail container services](#).

Step 2: Create a Dockerfile and build a container image

Complete the following procedure to create a Dockerfile, and build a `mystaticwebsite` Docker container image from it. The container image will be for a simple static website hosted on an Apache web server on Ubuntu.

1. Create a `mystaticwebsite` folder on your local machine where you will store your Dockerfile.
2. Create a Dockerfile in the folder you just created.

The Dockerfile does not use a file extension, such as `.TXT`. The full file name is `Dockerfile`.

3. Copy one of the following code blocks depending on how you want to configure your container image, and paste it into your Dockerfile:

- **If you want to create a simple static website container image with a Hello World message**, then copy the following code block and paste it into your Dockerfile. This code sample uses the Ubuntu 18.04 image. The RUN instructions updates the package caches, and installs and configures Apache, and prints a Hello World message to the web server's

document root. The EXPOSE instruction exposes port 80 on the container, and the CMD instruction starts the web server.

```
FROM ubuntu:18.04

# Install dependencies
RUN apt-get update && \
  apt-get -y install apache2

# Write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Open port 80
EXPOSE 80

# Start Apache service
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

- **If you want to use your own set of HTML files for your static website container image,** create an `html` folder in the same folder where you store your Dockerfile. Then put your HTML files in that folder.

After your HTML files are in the `html` folder, copy the following code block and paste into to your Dockerfile. This code sample uses the Ubuntu 18.04 image. The RUN instructions updates the package caches, and installs and configures Apache. The COPY instruction copies the contents of the `html` folder to the web server's document root. The EXPOSE instruction exposes port 80 on the container, and the CMD instruction starts the web server.

```
FROM ubuntu:18.04

# Install dependencies
RUN apt-get update && \
  apt-get -y install apache2

# Copy html directory files
COPY html /var/www/html/

# Open port 80
EXPOSE 80

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```


4. Open a command prompt or terminal window and change the directory to the folder in which you are storing your Dockerfile.
5. Enter the following command to build your container image using the Dockerfile in the folder. This command builds a new Docker container image named `mystaticwebsite`.

```
docker build -t mystaticwebsite .
```

You should see a message that confirms your image was successfully built.

6. Enter the following command to view the container images on your local machine.

```
docker images --filter reference=mystaticwebsite
```

You should see a result similar to the following example, showing the new container image created.

```
C:\Users\...Documents\Docker\Dockerfiles\mystaticwebsite>docker images --filter reference=mystaticwebsite
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
mystaticwebsite     latest      8f7ffd1013e0     8 minutes ago   199MB
```

Your newly built container image is ready to be tested by using it to run a new container on your local machine. Continue to the next [Step 3: Run your new container image](#) section of this guide.

Step 3: Run your new container image

Complete the following steps to run the new container image you created.

1. In a command prompt or terminal window, enter the following command to run the container image that you built in the previous [Step 2: Create a Dockerfile and build a container image](#) section of this guide. The `-p 8080:80` option maps the exposed port 80 on the container to port 8080 on your local machine. The `-d` option specifies that the container should run in detached mode.

```
docker container run -d -p 8080:80 --name mystaticwebsite mystaticwebsite:latest
```

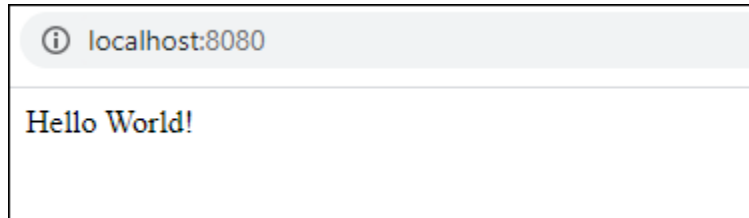
2. Enter the following command to view your running containers.

```
docker container ls -a
```

You should see a result similar to the following example, showing the new running container.

```
C:\Users\... \Documents\Docker\Dockerfiles\mystaticwebsite>docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
62382081e06b  mystaticwebsite:latest  "/bin/sh -c /root/ru..."  6 minutes ago  Up 6 minutes  0.0.0.0:8080->80/tcp    mystaticwebsite
```

- To confirm that the container is up and running, open a new browser window and browse to `http://localhost:8080`. You should see a message similar to the following example. This confirms that your container is up and running on your local machine.



Your newly built container image is ready to be pushed to your Lightsail account so that you can deploy it to your Lightsail container service. For more information, see [Pushing and managing container images on your Amazon Lightsail container services](#).

(Optional) Step 4: Clean up the containers running on your local machine

Now that you've created a container image that you can push to your Lightsail container service, it's time to clean up the containers that are running on your local machine as a result of following the procedures in this guide.

Complete the following steps to clean up the containers running on your local machine:

- Run the following command to view the containers that are running on your local machine.

```
docker container ls -a
```

You should see a result similar to the following, which lists the names of the containers running on your local machine.

```
C:\Users\... \Documents\Docker\Dockerfiles\mystaticwebsite>docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
62382081e06b  mystaticwebsite:latest  "/bin/sh -c /root/ru..."  6 minutes ago  Up 6 minutes  0.0.0.0:8080->80/tcp    mystaticwebsite
```

- Run the following command to remove the running container that you created earlier in this guide. This forces the container to be stopped, and permanently deletes it.

```
docker container rm <ContainerName> --force
```

In the command, replace <ContainerName> with the name of the container you want to stop, and delete.

Example:

```
docker container rm mystaticwebsite --force
```

The container that was created as a result of this guide should now be deleted.

Next steps after creating container images

After you create your container images, push them to your Lightsail container service when you're ready to deploy them. For more information, see [Manage Lightsail container service images](#).

Topics

- [Manage Lightsail container service images](#)
- [Install the Lightsail container services plugin](#)
- [Manage Amazon ECR private repository access in Lightsail](#)

Manage Lightsail container service images

When you create a deployment in your Amazon Lightsail container service, you must specify a source container image for each container entry. You can use images from a public registry, such as Amazon ECR Public Gallery, or you can use images that you create on your local machine. In this guide, we show you how to push container images from your local machine to your Lightsail container service. For more information about creating container images, see [Create container service images](#).

Contents

- [Prerequisites](#)
- [Push container images from your local machine to your container service](#)
- [View container images stored on your container service](#)

- [Delete container images stored on your container service](#)

Prerequisites

Complete the following prerequisites before you get started with pushing your container images to your container service:

- Create your container service in your Lightsail account. For more information, see [Creating Amazon Lightsail container services](#).
- Install software on your local machine that you need to create your own container images and push them to your Lightsail container service. For more information, see [Installing software to manage container images for your Amazon Lightsail container services](#).
- Create container images on your local machine, that you can push to your Lightsail container service. For more information, see [Creating container images for your Amazon Lightsail container services](#).

Push container images from your local machine to your container service

Complete the following procedure to push your container images to your container service.

1. Open a command prompt or terminal window.
2. In the command prompt or terminal window, enter the following command to view the Docker images that are currently on your local machine.

```
docker images
```

3. In the result, locate the name (repository name) and tag of the container image that you want to push to your container service. Make a note of it because you will need it in the next step.

```
C:\WINDOWS\system32>docker images
REPOSITORY          TAG                IMAGE ID           CREATED            SIZE
mystaticwebsite     v2                cd5f05cb6ddf      33 minutes ago    188MB
mystaticwebsite     v1                9c7d52450629      3 hours ago       188MB
```

4. Enter the following command to push the container image on your local machine to your container service.

```
aws lightsail push-container-image --region <Region> --service-  
name <ContainerServiceName> --label <ContainerImageLabel> --  
image <LocalContainerImageName>:<ImageTag>
```

In the command, replace:

- *<Region>* with the AWS Region in which your container service was created.
- *<ContainerServiceName>* with the name of your container service.
- *<ContainerImageLabel>* with the label that you want to give your container image when it's stored on your container service. Specify a descriptive label that you can use to track the different versions of your registered container images.

The label will be part of the container image name generated by your container service. For example, if your container service name is `container-service-1`, the container image label is `mystaticsite`, and this is the first version of the container image you're pushing, then the image name generated by your container service will be `:container-service-1.mystaticsite.1`.

- *<LocalContainerImageName>* with the name of the container image that you want to push to your container service. You obtained the container image name in the previous step of this procedure.
- *<ImageTag>* with the tag of the container image that you want to push to your container service. You obtained the container image tag in the previous step of this procedure.

Example:

```
aws lightsail push-container-image --region us-west-2 --service-name myservice --  
label mystaticwebsite --image mystaticwebsite:v2
```

You should see a result similar to the following example, which confirms that your container image was pushed to your container service.

```
C:\WINDOWS\system32>aws lightsail push-container-image --service-name myservice --label mystaticwebsite
--image mystaticwebsite:v2

[185a355b95: Preparing
[180994b087: Preparing
[180c904ff3: Preparing
[18370aa736: Preparing
[18f192bbc8: Preparing
[18bc0bd923: Preparing
[7BDigest: sha256:3a585ca39bba342e390b39f2fea00bbc20f492c0cda7b923dd766abe31918f3b8/1.96kB
Image "mystaticwebsite:v2" registered.
Refer to this image as ":myservice.mystaticwebsite.2" in deployments.
```

Refer to the following [View container images stored on your container service](#) section of this guide to view your pushed container image in your container service on the Lightsail console.

View container images stored on your container service

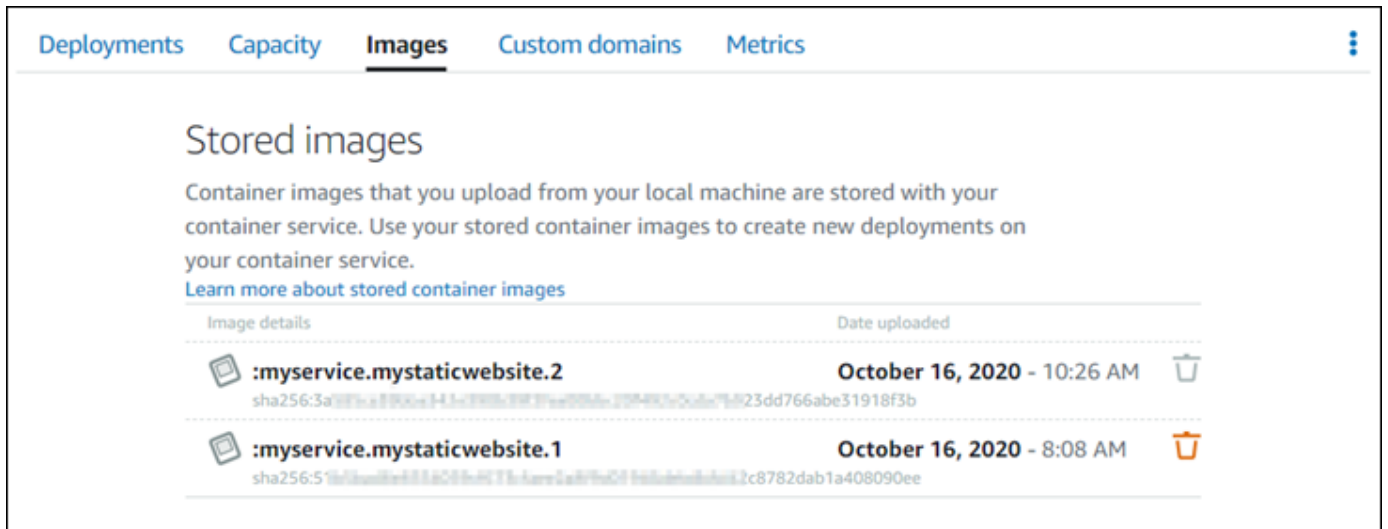
Complete the following procedure to view container images that were pushed, and are being stored, on your container service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to view the stored container images.
4. On the container service management page, choose the **Images** tab.

Note

The **Images** tab is not displayed if you have not pushed images to your container service. To display the images tab for your container service you must first push container images to your service.

The **Images** page lists the container images that were pushed to your container service, and are currently being stored on your service. Container images that are being used in a current deployment cannot be deleted and are listed with a grayed-out delete icon.



You can create deployments using container images stored on your service. For more information, see [Creating and managing deployments for your Amazon Lightsail container services](#).

Delete container images stored on your container service

Complete the following procedure to delete container images that were pushed, and are being stored, on your container service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to view the current deployment.
4. On the container service management page, choose the **Images** tab.

Note

The **Images** tab is not displayed if you have not pushed images to your container service. To display the images tab for your container service you must first push container images to your service.

5. Find the container image you want to delete, and choose the delete (trash bin) icon.

Note

Container images that are being used in a current deployment cannot be deleted and their delete icons are grayed-out.

6. In the confirmation prompt that appears, choose **Yes, delete** to confirm that you want to permanently delete the stored image.

Your stored container image is immediately deleted from your container service.

Install the Lightsail container services plugin

You can use the Amazon Lightsail console to create your Lightsail container services, and create deployments using container images from an online public registry, such as Amazon ECR Public Gallery. To create your own container images, and push them to your container service, you must install the following additional software on the same computer on which you plan to create your container images:

- **Docker** – Run, test, and create your own container images that you can then use with your Lightsail container service.
- **AWS Command Line Interface (AWS CLI)** – Specify parameters of the container images you create, and then push them to your Lightsail container service. Version 2.1.1 and later will work with the Lightsail Control plugin.
- **Lightsail Control (lightsailctl) plugin** – Enables the AWS CLI to access the container images that are on the local machine.

The following sections of this guide describe where to go to download these software packages, and how to install them. For more information about container services, see [Container services](#).

Contents

- [Install Docker](#)
- [Install the AWS CLI](#)
- [Install the Lightsail Control plugin](#)
 - [Install the lightsailctl plugin on Windows](#)
 - [Install the lightsailctl plugin on macOS](#)

- [Install the lightsailctl plugin on Linux](#)

Install Docker

Docker is a technology that allows you to build, run, test, and deploy distributed applications that are based on Linux containers. You must install and use Docker software if you want to create your own container images that you can then use with your Lightsail container service. For more information, see [Create container images for your Lightsail container services](#).

Docker is available for many different operating systems, including most modern Linux distributions, like Ubuntu, and even macOS and Windows. For more information about how to install Docker on your particular operating system, see the [Docker installation guide](#).

Note

Always install the latest version of Docker. Older versions of Docker are not guaranteed to work with the AWS CLI and Lightsail Control (lightsailctl) plugin described later in this guide.

Install the AWS CLI

The AWS CLI is an open source tool that enables you to interact with AWS services, such as Lightsail, using commands in your command-line shell. You must install and use the AWS CLI to push your container images, created on your local machine, to your Lightsail container service.

The AWS CLI is available in the following versions:

- **Version 2.x** – The current, generally available release of the AWS CLI. This is the most recent major version of the AWS CLI and supports all of the latest features, including the ability to push container images to your Lightsail container services. Version 2.1.1 and later will work with the Lightsail Control plugin.
- **Version 1.x** – The previous version of the AWS CLI that is available for backwards compatibility. This version does not support the ability to push your container images to your Lightsail container services. Therefore, you must install the AWS CLI version 2 instead.

The AWS CLI version 2 is available for Linux, macOS, and Windows operating systems. For instructions on how to install the AWS CLI on those operating systems, see [Installing the AWS CLI version 2](#) in the *AWS CLI User Guide*.

Install the Lightsail Control plugin

The Lightsail Control (`lightsailctl`) plugin is a lightweight application that allows the AWS CLI to access the container images that you created on your local machine. It allows you to push container images to your Lightsail container service, so that you can deploy them to your service.

System requirements

- A Windows, macOS, or Linux operating system with 64-bit support.
- AWS CLI version 2 must be installed on your local machine in order to use the `lightsailctl` plugin. For more information, see the [Install the AWS CLI](#) section earlier in this guide.

Use the latest version of the `lightsailctl` plugin

The `lightsailctl` plugin is updated occasionally with enhanced functionality. Each time you use the `lightsailctl` plugin, it performs a check to confirm you're using the latest version. If it finds that a new version is available, it prompts you to update to the latest version to take advantage of the latest features. When an updated version is available, you must repeat the installation process to get the latest version of the `lightsailctl` plugin.

The following lists all releases of the `lightsailctl` plugin and the features and enhancements included with each version.

- **v1.0.0 (released November 12, 2020)** – Initial release adds functionality for the AWS CLI version 2 to push container images to a Lightsail container service.

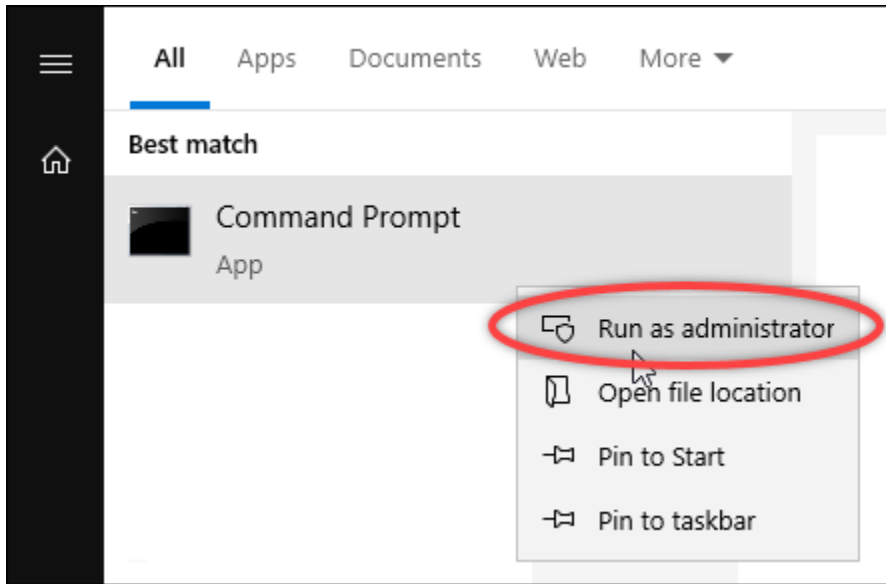
Install the `lightsailctl` plugin on Windows

Complete the following procedure to install the `lightsailctl` plugin on Windows.

1. Download the executable from the following URL, and save it to the `C:\Temp\lightsailctl\` directory.

```
https://s3.us-west-2.amazonaws.com/lightsailctl/latest/windows-amd64/lightsailctl.exe
```

2. Choose the **Windows Start** button, and then search for cmd.
3. Right-click the **Command Prompt** application in the results, and choose **Run as administrator**.



Note

You may see a prompt that asks if you want to allow Command Prompt to make changes to your device. You must choose **Yes** to continue with the installation.

4. Enter the following command to set a path environment variable that points to the C:\Temp\lightsailctl\ directory where you saved the lightsailctl plugin.

```
setx PATH "%PATH%;C:\Temp\lightsailctl" /M
```

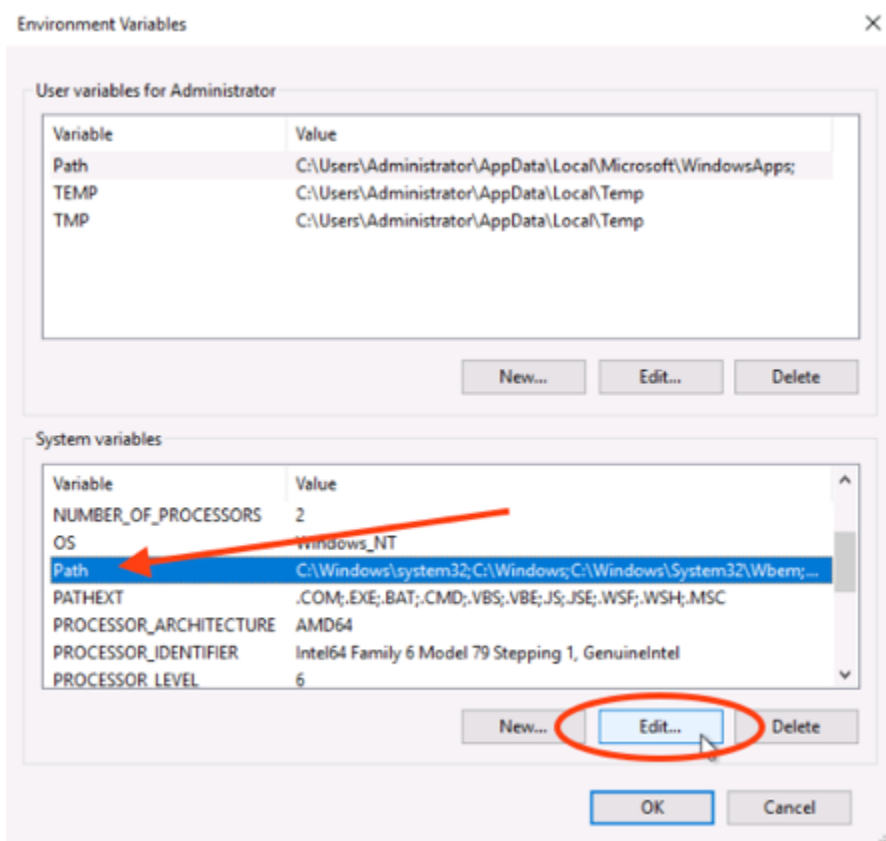
You should see a result similar to the following example.

```
C:\WINDOWS\system32>setx PATH "%PATH%;C:\Temp\lightsailctl\" /M
SUCCESS: Specified value was saved.
```

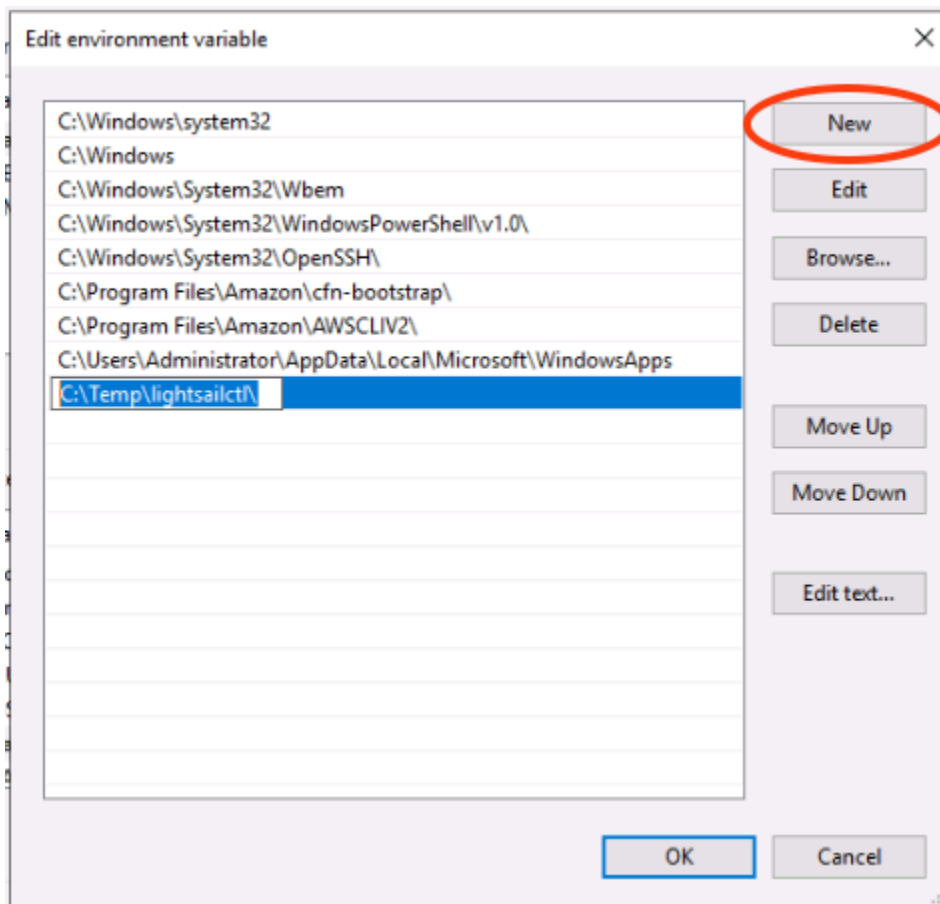
The setx command will truncate beyond 1024 characters. Use the following procedure to manually set the path environment variable if you already have multiple variables set in your PATH.

1. On the **Start** menu, open **Control Panel**.
2. Choose **System and Security**, then **System**.

3. Choose **Advanced system settings**.
4. On the **Advanced** tab of the **System Properties** dialog box, choose **Environment Variables**.
5. In the **System Variables** box of the **Environment Variables** dialog box, select **Path**.
6. Choose the **Edit** button located under the **System Variables** box.



7. Choose **New**, then enter the following path: `C:\Temp\lightsailctl\`



8. Choose **OK** in three successive dialog boxes, and then close the **System** dialog box.

You are now ready to use the AWS Command Line Interface (AWS CLI) to push container images to your Lightsail container service. For more information, see [Push and manage container images](#).

Install the lightsailctl plugin on macOS

Complete one of the following procedures to download and install the lightsailctl plugin on macOS.

Homebrew download and install

1. Open a terminal window.
2. Enter the following command to download and install the lightsailctl plugin.

```
brew install aws/tap/lightsailctl
```

Note

For more information about Homebrew, see the [Homebrew](#) website.

Manual download and install

1. Open a terminal window.
2. Enter the following command to download the lightsailctl plugin and copy it to the bin folder.

```
curl "https://s3.us-west-2.amazonaws.com/lightsailctl/latest/darwin-amd64/lightsailctl" -o "/usr/local/bin/lightsailctl"
```

3. Enter the following command to make the plugin executable.

```
chmod +x /usr/local/bin/lightsailctl
```

4. Enter the following command to clear extended attributes for the plugin.

```
xattr -c /usr/local/bin/lightsailctl
```

You are now ready to use the AWS CLI to push container images to your Lightsail container service. For more information, see [Push and manage container images](#).

Install the lightsailctl plugin on Linux

Complete the following procedure to install the Lightsail container services plugin on Linux.

1. Open a terminal window.
2. Enter the following command to download the lightsailctl plugin.
 - For the AMD 64-bit architecture version of the plugin:

```
curl "https://s3.us-west-2.amazonaws.com/lightsailctl/latest/linux-amd64/lightsailctl" -o "/usr/local/bin/lightsailctl"
```

- For the ARM 64-bit architecture version of the plugin:

```
curl "https://s3.us-west-2.amazonaws.com/lightsailctl/latest/linux-arm64/lightsailctl" -o "/usr/local/bin/lightsailctl"
```

3. Enter the following command to make the plugin executable.

```
sudo chmod +x /usr/local/bin/lightsailctl
```

You are now ready to use the AWS CLI to push container images to your Lightsail container service. For more information, see [Push and manage container images](#).

Manage Amazon ECR private repository access in Lightsail

Amazon Elastic Container Registry (Amazon ECR) is an AWS managed container image registry service that supports private repositories with resource-based permissions using AWS Identity and Access Management (IAM). You can give your Amazon Lightsail container services access to your Amazon ECR private repositories. Then, you can deploy images from your private repository to your container services.

You can manage access for your Lightsail container services and your Amazon ECR private repositories by using the Lightsail console or the AWS Command Line Interface (AWS CLI). However, we recommend that you use the Lightsail console because it simplifies the process.

For more information about container services, see [Container services](#). For more information about Amazon ECR, see the [Amazon ECR User Guide](#).

Contents

- [Required permissions](#)
- [Use the Lightsail console to manage access to private repositories](#)
- [Use the AWS CLI to manage access to private repositories](#)
 - [Activate or deactivate the Amazon ECR image puller IAM role](#)
 - [Determine if your Amazon ECR private repository has a policy statement](#)
 - [Add a policy to a private repository that doesn't have a policy statement](#)
 - [Add a policy to a private repository that has a policy statement](#)

Required permissions

The user who will manage access for Lightsail container services to Amazon ECR private repositories must have one of the following permissions policies in IAM. For more information, see [Adding and removing IAM identity permissions](#) in the *AWS Identity and Access Management User Guide*.

Grant access to any Amazon ECR private repository

The following permissions policy grants a user permission to configure access to any Amazon ECR private repository.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ManageEcrPrivateRepositoriesAccess",
      "Effect": "Allow",
      "Action": [
        "ecr:SetRepositoryPolicy",
        "ecr:DescribeRepositories",
        "ecr>DeleteRepositoryPolicy",
        "ecr:GetRepositoryPolicy"
      ],
      "Resource": "arn:aws:ecr:*:AwsAccountId:repository/*"
    }
  ]
}
```

In the policy, replace *AwsAccountId* with your AWS account ID number.

Grant access to a specific Amazon ECR private repository

The following permissions policy grants a user permission to configure access to a specific Amazon ECR private repository, in a specific AWS Region.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ManageEcrPrivateRepositoriesAccess",
      "Effect": "Allow",
      "Action": [
```



```

        "ecr:SetRepositoryPolicy",
        "ecr:DescribeRepositories",
        "ecr>DeleteRepositoryPolicy",
        "ecr:GetRepositoryPolicy"
    ],
    "Resource": "arn:aws:ecr:AwsRegion:AwsAccountId:repository/RepositoryName"
}
]
}

```

In the policy, replace the following example text with your own:

- *AwsRegion* — The AWS Region code (for example, us-east-1) of the private repository. Your Lightsail container service must be in the same AWS Region as the private repositories that you want to access.
- *AwsAccountId* — Your AWS account ID number.
- *RepositoryName* — The name of the private repository for which you want to manage access.

Following is an example of the permissions policy populated with example values.

```

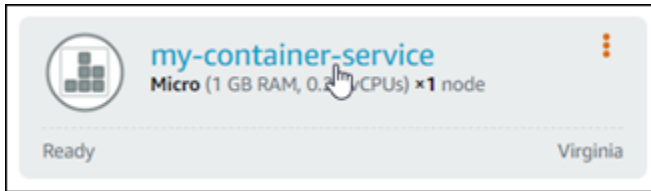
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ManageEcrPrivateRepositoriesAccess",
      "Effect": "Allow",
      "Action": [
        "ecr:SetRepositoryPolicy",
        "ecr:DescribeRepositories",
        "ecr>DeleteRepositoryPolicy",
        "ecr:GetRepositoryPolicy"
      ],
      "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/my-private-repo"
    }
  ]
}

```

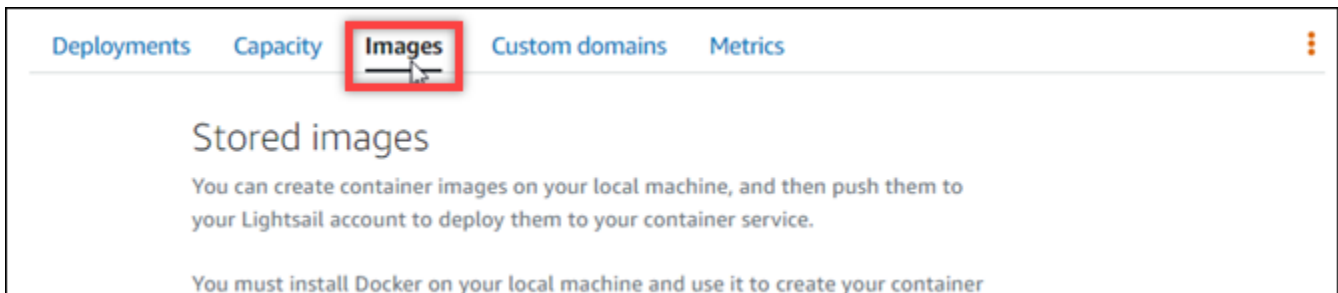
Use the Lightsail console to manage access to private repositories

Complete the following procedure to use the Lightsail console to manage access for a Lightsail container service to an Amazon ECR private repository.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to configure access to an Amazon ECR private repository.



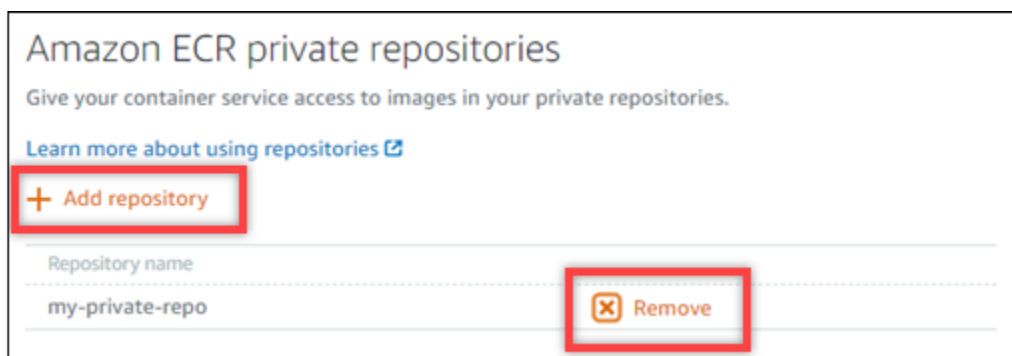
4. Choose the **Images** tab.



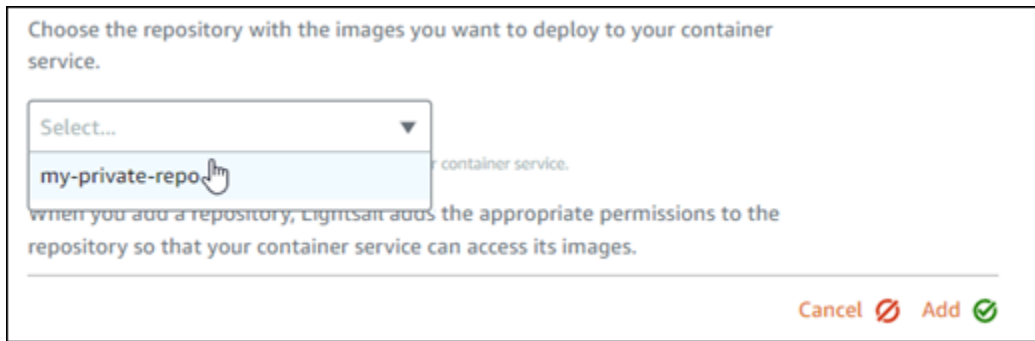
5. Choose **Add repository** to grant access for your container service to an Amazon ECR private repository.

Note

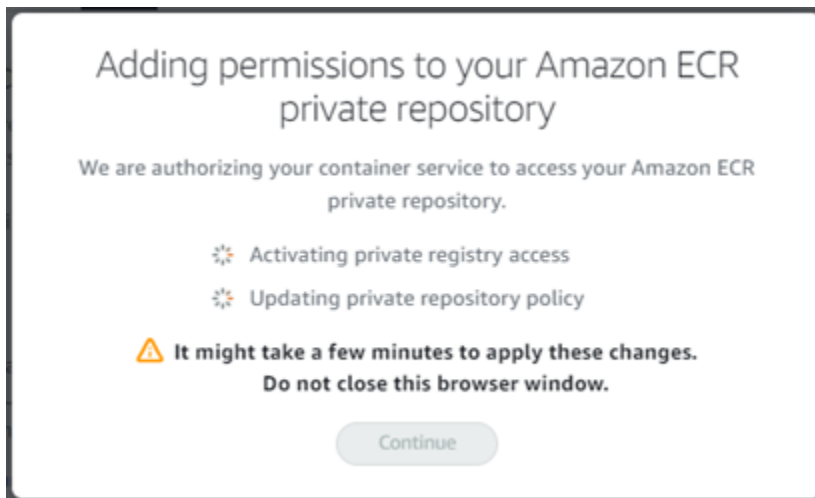
You can choose **Remove** to remove access for your container service from a previously added Amazon ECR private repository.



6. In the dropdown that appears, select the private repository that you would like to access, and then choose **Add**.



Lightsail takes a few moments to activate the Amazon ECR image puller IAM role for your container service, which includes a principal Amazon Resource Name (ARN). Lightsail then automatically adds the IAM role principal ARN to the permissions policy of the Amazon ECR private repository that you selected. This grants your container service access to the private repository and its images. Don't close the browser window until the modal that appears indicates that the process is completed and you can choose **Continue**.



7. Choose **Continue** when the activation is completed.

After the selected Amazon ECR private repository is added it is listed in the **Amazon ECR private repositories** section of the page. The page includes instructions for how to deploy an image from the private repository to your Lightsail container service. To use an image from your private repository, specify the URI format that is displayed on the page as the **Image** value when creating your container service deployment. In the URI that you specify, replace the example `{image tag}` with the tag of the image you want to deploy. For more information, see [Create and manage container service deployments](#).

Next steps

To deploy an image from your private repository, configure a container service deployment with the following URI format in the image field:

```
111122223333.dkr.ecr.us-east-1.amazonaws.com/my-private-repo:{image tag}
```

You can manage your private repositories and images using the Amazon ECR console.

[Open the Amazon ECR console](#)

Use the AWS CLI to manage access to private repositories

Managing access for a Lightsail container service to an Amazon ECR private repository using the AWS Command Line Interface (AWS CLI) requires the following steps:

Important

We recommend that you use the Lightsail console to manage access for a Lightsail container service to an Amazon ECR private repository because it simplifies the process. For more information, see [Use the Lightsail console to manage access to private repositories](#) earlier in this guide.

1. **Activate or deactivate the Amazon ECR image puller IAM role** — Use the AWS CLI `update-container-service` command for Lightsail to activate or deactivate the Amazon ECR image puller IAM role. A principal Amazon Resource Name (ARN) is created for the Amazon ECR image puller IAM role when you activate it. For more information, see the [Activate or deactivate the Amazon ECR image puller IAM role](#) section of this guide.
2. **Determine if your Amazon ECR private repository has a policy statement** — After you activate the Amazon ECR image puller IAM role, you need to determine if the Amazon ECR private repository that you want to access with your container service has an existing policy statement. For more information, see [Determine if your Amazon ECR private repository has a policy statement](#) later in this guide.

You add the IAM role principal ARN to your repository using one of the following methods, depending on whether your repository has an existing policy statement:

- a. **Add a policy to a private repository that doesn't have a policy statement** — Use the AWS CLI `set-repository-policy` command for Amazon ECR to add the Amazon ECR image

puller role principal ARN for your container service to a private repository that has an existing policy. For more information, see [Add a policy to a private repository that doesn't have a policy statement](#) later in this guide.

- b. **Add a policy to a private repository that has a policy statement** — Use the AWS CLI `set-registry-policy` command for Amazon ECR to add the Amazon ECR image puller role for your container service to a private repository that doesn't have an existing policy. For more information, see [Add a policy to a private repository that has a policy statement](#) later in this guide.

Activate or deactivate the Amazon ECR image puller IAM role

Complete the following procedure to activate or deactivate the Amazon ECR image puller IAM role for your Lightsail container service. You can activate or deactivate the Amazon ECR image puller IAM role using the AWS CLI `update-container-service` command for Lightsail. For more information, see [update-container-service](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail before you can continue with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to update a container service and activate or deactivate the Amazon ECR image puller IAM role.

```
aws lightsail update-container-service --service-name ContainerServiceName --  
private-registry-access ecrImagePullerRole={isActive=RoleActivationState} --  
region AwsRegionCode
```

In the command, replace the following example text with your own:

- *ContainerServiceName* — The name of the container service for which to activate or deactivate the Amazon ECR image puller IAM role.
- *RoleActivationState* — The activation state of the Amazon ECR image puller IAM role. Specify `true` to activate the role, or `false` to deactivate it.

- *AwsRegionCode* — The AWS Region code of the container service (for example, `us-east-1`).

Examples:

- To activate the Amazon ECR image puller IAM role:

```
aws lightsail update-container-service --service-name my-container-service --private-registry-access ecrImagePullerRole={isActive=true} --region us-east-1
```

- To deactivate the Amazon ECR image puller IAM role:

```
aws lightsail update-container-service --service-name my-container-service --private-registry-access ecrImagePullerRole={isActive=false} --region us-east-1
```

3. If you:

- **Activated the Amazon ECR image puller role** — Wait at least 30 seconds after getting the previous response. Then, continue to the next step to get the principal ARN of the Amazon ECR image puller IAM role for your container service.
- **Deactivated the Amazon ECR image puller role** — If you previously added the Amazon ECR image puller IAM role principal ARN to the permissions policy of your Amazon ECR private repository, you should remove that permissions policy from your repository. For more information, see [Deleting a private repository policy statement](#) in the *Amazon ECR User Guide*.

4. Enter the following command to get the principal ARN of the Amazon ECR image puller IAM role for your container service.

```
aws lightsail get-container-services --service-name ContainerServiceName --region AwsRegionCode
```

In the command, replace the following example text with your own:

- *ContainerServiceName* — The name of your container service for which to get the Amazon ECR image puller IAM role principal ARN.
- *AwsRegionCode* — The AWS Region code of the container service (for example, `us-east-1`).

Example:

```
aws lightsail get-container-services --service-name my-container-service --  
region us-east-1
```

Look for the ECR image puller IAM role principal ARN in the response. If a role is listed, copy it or write it down. You will need it for the next section of this guide. Next, you need to determine if there is an existing policy statement on the Amazon ECR private repository that you want to access with your container service. Continue to the [Determine if your Amazon ECR private repository has a policy statement](#) section of this guide.

Determine if your Amazon ECR private repository has a policy statement

Use the following procedure to determine if your Amazon ECR private repository has a policy statement. You can use the AWS CLI `get-repository-policy` command for Amazon ECR. For more information, see [update-container-service](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Amazon ECR before you can continue with this procedure. For more information, see [Setting up with Amazon ECR](#) in the *Amazon ECR User Guide*.

1. Open a Command Prompt or Terminal window.
2. Enter the following command to get the policy statement for a specific private repository.

```
aws ecr get-repository-policy --repository-name RepositoryName --  
region AwsRegionCode
```

In the command, replace the following example text with your own:

- *RepositoryName* — The name of the private repository for which you want to configure access for a Lightsail container service.
- *AwsRegionCode* — The AWS Region code of the private repository (for example, `us-east-1`).

Example:

```
aws ecr get-repository-policy --repository-name my-private-repo --region us-east-1
```

You should see one of the following responses:

- **RepositoryPolicyNotFoundException** — Your private repository does not have a policy statement. If your repository doesn't have a policy statement, follow the steps in the [Add a policy to a private repository that doesn't have a policy statement](#) section later in this guide.

```
C:\>aws ecr get-repository-policy --repository-name my-private-repo

An error occurred (RepositoryPolicyNotFoundException) when calling the GetRepositoryPolicy operation: Repository policy does not exist for the repository with name 'my-private-repo' in the registry with id '12345678901'
```

- **A repository policy was found** - Your private repository has a policy statement, and it is displayed in the response of your request. If your repository has a policy statement, copy the existing policy and then follow the steps in the [Add a policy to a private repository that has a policy statement](#) section later in this guide.

```
C:\>aws ecr get-repository-policy --repository-name my-private-repo
{
  "registryId": "12345678901",
  "repositoryName": "my-private-repo",
  "policyText": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement\" : [ {\n    \"Sid\" : \"AllowUserPushPull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" : {\n      \"AWS\" : \"arn:aws:iam::12345678901:user/example-user\"\n    },\n    \"Action\" : [ \"ecr:BatchGetImage\", \"ecr:BatchCheckLayerAvailability\", \"ecr:CompleteLayerUpload\", \"ecr:GetDownloadUrlForLayer\", \"ecr:InitiateLayerUpload\", \"ecr:PutImage\", \"ecr:UploadLayerPart\" ]\n  } ]\n}"
```

Add a policy to a private repository that doesn't have a policy statement

Complete the following procedure to add a policy to an Amazon ECR private repository that doesn't have a policy statement. The policy that you add must include the Amazon ECR image puller IAM role principal ARN of your Lightsail container service. This grants access for your container service to deploy images from the private repository.

Important

Lightsail automatically adds the Amazon ECR image puller role to your Amazon ECR private repositories when you use the Lightsail console to configure access. In that case, you don't have to manually add the Amazon ECR image puller role to your private repositories

using the procedure in this section. For more information, see [Use the Lightsail console to manage access to private repositories](#) earlier in this guide.

You can add a policy to a private repository using the AWS CLI. You do this by creating a JSON file that contains the policy, and then referencing that file with the `set-repository-policy` command for Amazon ECR. For more information, see [set-repository-policy](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Amazon ECR before continuing with this procedure. For more information, see [Setting up with Amazon ECR](#) in the *Amazon ECR User Guide*.

1. Open a text editor, and paste the following policy statement into a new text file.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AllowLightsailPull-ecr-private-repo-demo",
      "Effect": "Allow",
      "Principal": {
        "AWS": "IamRolePrincipalArn"
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
      ]
    }
  ]
}
```

In the text, replace *IamRolePrincipalArn* with the Amazon ECR image puller IAM role principal ARN of your container service that you got earlier in this guide.

2. Save the file as `ecr-policy.json` to an accessible location on your computer (for example, `C:\Temp\ecr-policy.json` on Windows or `/tmp/ecr-policy.json` on macOS or Linux).

3. Write down the file path location of the `ecr-policy.json` file created. You will specify it in a command later in this procedure.
4. Open a Command Prompt or Terminal window.
5. Enter the following command to set the policy statement for the private repository that you want to access with your container service.

```
aws ecr set-repository-policy --repository-name RepositoryName --policy-text  
file://path/to/ecr-policy.json --region AwsRegionCode
```

In the command, replace the following example text with your own:

- *RepositoryName* — The name of the private repository for which you want to add the policy.
- *path/to/* — The path to the `ecr-policy.json` file on your computer that you created earlier in this guide.
- *AwsRegionCode* — The AWS Region code of the private repository (for example, `us-east-1`).

Examples:

- On Windows:

```
aws ecr set-repository-policy --repository-name my-private-repo --policy-text  
file://C:\Temp\ecr-policy.json --region us-east-1
```

- On macOS or Linux:

```
aws ecr set-repository-policy --repository-name my-private-repo --policy-text  
file:///tmp/ecr-policy.json --region us-east-1
```

Your container service is now able to access your private repository and its images. To use an image from your repository, specify the following URI as the **Image** value for your container service deployment. In the URI, replace the example *tag* with the tag of the image you want to deploy. For more information, see [Create and manage container service deployments](#).

```
AwsAccountId.dkr.ecr.AwsRegionCode.amazonaws.com/RepositoryName:ImageTag
```

In the URI, replace the following example text with your own:

- *AwsAccountId* — Your AWS account ID number.
- *AwsRegionCode* — The AWS Region code of the private repository (for example, us-east-1).
- *RepositoryName* — The name of the private repository from which to deploy a container image.
- *ImageTag* — The tag of the container image from the private repository to deploy on your container service.

Example:

```
111122223333.dkr.ecr.us-east-1.amazonaws.com/my-private-repo:myappimage
```

Add a policy to a private repository that has a policy statement

Complete the following procedure to add a policy to an Amazon ECR private repository that has a policy statement. The policy that you add must include the existing policy and a new policy that contains the Amazon ECR image puller IAM role principal ARN of your Lightsail container service. This maintains the existing permissions on your private repository while also granting access for your container service to deploy images from the private repository.

Important

Lightsail automatically adds the Amazon ECR image puller role to your Amazon ECR private repositories when you use the Lightsail console to configure access. In that case, you don't have to manually add the Amazon ECR image puller role to your private repositories using the procedure in this section. For more information, see [Use the Lightsail console to manage access to private repositories](#) earlier in this guide.

You can add a policy to a private repository using the AWS CLI. You do this by creating a JSON file that contains the existing policy and the new policy. Then, reference that file with the `set-repository-policy` command for Amazon ECR. For more information, see [set-repository-policy](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Amazon ECR before you can continue with this procedure. For more information, see [Setting up with Amazon ECR](#) in the *Amazon ECR User Guide*.

1. Open a Command Prompt or Terminal window.
2. Enter the following command to get the policy statement for a specific private repository.

```
aws ecr get-repository-policy --repository-name RepositoryName --
region AwsRegionCode
```

In the command, replace the following example text with your own:

- *RepositoryName* — The name of the private repository for which you want to configure access for a Lightsail container service.
- *AwsRegionCode* — The AWS Region code of the private repository (for example, us-east-1).

Example:

```
aws ecr get-repository-policy --repository-name my-private-repo --region us-east-1
```

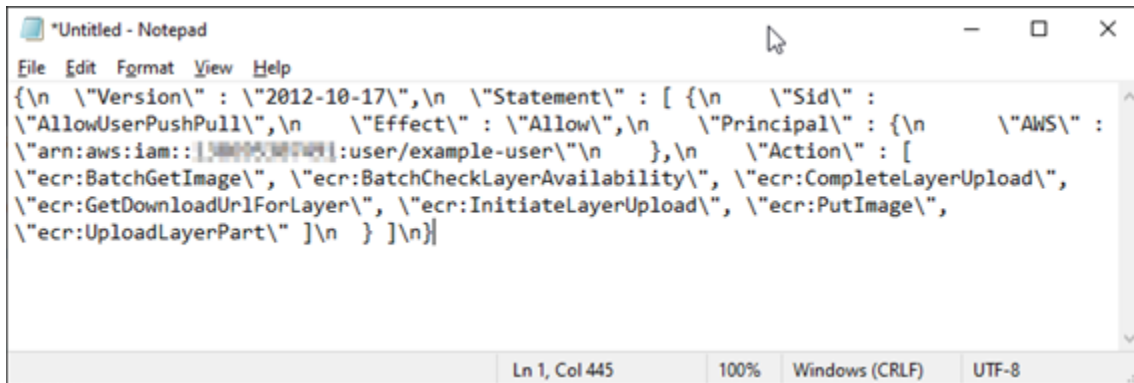
3. In the response, copy the existing policy and continue to the next step.

You should copy only the content of the `policyText` that appears between the double quotes, as highlighted in the following example.

```
C:\>aws ecr get-repository-policy --repository-name my-private-repo
{
  "registryId": "111111111111",
  "repositoryName": "my-private-repo",
  "policyText": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n    {\n      \"Sid\": \"AllowUserPushPull\",\n      \"Effect\": \"Allow\",\n      \"Principal\": {\n        \"AWS\": \"arn:aws:iam::111111111111:user/example-user\"\n      },\n      \"Action\": [\n        \"ecr:BatchGetImage\",\n        \"ecr:BatchCheckLayerAvailability\",\n        \"ecr:CompleteLayerUpload\",\n        \"ecr:GetDownloadUrlForLayer\",\n        \"ecr:InitiateLayerUpload\",\n        \"ecr:PutImage\",\n        \"ecr:UploadLayerPart\"\n      ]\n    }\n  ]\n}"
```

4. Open a text editor, and paste the existing policy from your private repository that you copied in the previous step.

The result should look like the following example.



```

*Untitled - Notepad
File Edit Format View Help
{\n  \Version\" : \"2012-10-17\", \n  \Statement\" : [ {\n    \Sid\" :
\AllowUserPushPull\", \n    \Effect\" : \Allow\", \n    \Principal\" : {\n      \AWS\" :
\arn:aws:iam::1:111111111111:user/example-user\" \n    }, \n    \Action\" : [
\ecr:BatchGetImage\", \ecr:BatchCheckLayerAvailability\", \ecr:CompleteLayerUpload\",
\ecr:GetDownloadUrlForLayer\", \ecr:InitiateLayerUpload\", \ecr:PutImage\",
\ecr:UploadLayerPart\" ] \n  } ] \n}
Ln 1, Col 445    100%    Windows (CRLF)    UTF-8

```

5. In the text that you pasted, replace `\n` with line breaks and delete the remaining `\`.

The result should look like the following example.



```

*Untitled - Notepad
File Edit Format View Help
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPushPull",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::1:111111111111:user/example-user"
        ]
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability",
        "ecr:CompleteLayerUpload",
        "ecr:GetDownloadUrlForLayer",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ]
    }
  ]
}
Ln 23, Col 2    100%    Windows (CRLF)    UTF-8

```

6. Paste the following policy statement at the end of the text file.

```

{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AllowLightsailPull-ecr-private-repo-demo",
      "Effect": "Allow",

```

```
    "Principal": {
      "AWS": "IamRolePrincipalArn"
    },
    "Action": [
      "ecr:BatchGetImage",
      "ecr:GetDownloadUrlForLayer"
    ]
  }
]
```

7. In the text, replace *IamRolePrincipalArn* with the Amazon ECR image puller IAM role principal ARN of your container service that you got earlier in this guide.

The result should look like the following example.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPushPull",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/example-user"
        ]
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability",
        "ecr:CompleteLayerUpload",
        "ecr:GetDownloadUrlForLayer",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ]
    }
  ]
},
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AllowLightsailPull-ecr-private-repo-demo",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/amazon/lightsail/us-east-a/containers/my-container-service/private-repo-access/3EXAMPLEm8gmrcs1vEXAMPLEkkemufe7ime26fo9i7e5ct93k7ng"
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
      ]
    }
  ]
}

```

8. Save the file as `ecr-policy.json` to an accessible location on your computer (for example, `C:\Temp\ecr-policy.json` on Windows or `/tmp/ecr-policy.json` on macOS or Linux).
9. Write down the file path location of the `ecr-policy.json` file. You will specify it in a command later in this procedure.
10. Open a Command Prompt or Terminal window.
11. Enter the following command to set the policy statement for the private repository that you want to access with your container service.

```
aws ecr set-repository-policy --repository-name RepositoryName --policy-text
file://path/to/ecr-policy.json --region AwsRegionCode
```

In the command, replace the following example text with your own:

- *RepositoryName* — The name of the private repository for which you want to add the policy.
- *path/to/* — The path to the `ecr-policy.json` file on your computer that you created earlier in this guide.
- *AwsRegionCode* — The AWS Region code of the private repository (for example, `us-east-1`).

Examples:

- On Windows:

```
aws ecr set-repository-policy --repository-name my-private-repo --policy-text
file:///C:\Temp\ecr-policy.json --region us-east-1
```

- On macOS or Linux:

```
aws ecr set-repository-policy --repository-name my-private-repo --policy-text
file:///tmp/ecr-policy.json --region us-east-1
```

You should see a response similar to the following example.

```
C:\>aws ecr set-repository-policy --repository-name my-private-repo --policy-text file:///C:\Temp\ecr-policy.json --region
us-west-2
{
  "registryId": "123456789012",
  "repositoryName": "my-private-repo",
  "policyText": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n    {\n      \"Sid\": \"AllowLightsailPull-my-cont
ainer-service\",\n      \"Effect\": \"Allow\",\n      \"Principal\": {\n        \"AWS\": \"arn:aws:iam::123456789012:role/a
mazon/lightsail/us-west-2/containers/my-container-service/private-repo-access/iamrolelightsail-ecr-pull-access\"\n      },\n      \"Action\": [ \"ecr:BatchGetImage\", \"ecr:GetDownloadUrlForLayer\" ]\n    }, {\n      \"Sid\":
\"AllowUserPushPull\",\n      \"Effect\": \"Allow\",\n      \"Principal\": {\n        \"AWS\": \"arn:aws:iam::123456789012:role/
user/example-user\"\n      },\n      \"Action\": [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\", \"ecr:Comple
teLayerUpload\", \"ecr:GetDownloadUrlForLayer\", \"ecr:InitiateLayerUpload\", \"ecr:PutImage\", \"ecr:UploadLayerPart\"
 ]\n    } ]\n}"
```

If you run the `get-repository-policy` command again, you should see the new additional policy statement on your private repository. Your container service is now able to access your

private repository and its images. To use an image from your repository, specify the following URI as the **Image** value for your container service deployment. In the URI, replace the example *tag* with the tag of the image you want to deploy. For more information, see [Create and manage container service deployments](#).

```
AwsAccountId.dkr.ecr.AwsRegionCode.amazonaws.com/RepositoryName:ImageTag
```

In the URI, replace the following example text with your own:

- *AwsAccountId* — Your AWS account ID number.
- *AwsRegionCode* — The AWS Region code of the private repository (for example, us-east-1).
- *RepositoryName* — The name of the private repository from which to deploy a container image.
- *ImageTag* — The tag of the container image from the private repository to deploy on your container service.

Example:

```
111122223333.dkr.ecr.us-east-1.amazonaws.com/my-private-repo:myappimage
```

Create and manage container service deployments in Lightsail

Create a deployment when you're ready to launch containers on your Amazon Lightsail container service. A deployment is a set of specifications for the containers that you wish to launch on your service. Your container service can have one running deployment at a time, and a deployment can have up to 10 container entries. You can create a deployment at the same time as you create your container service, or you can create it after your service is up and running.

Note

If you create a new deployment, then the existing utilization metrics of your container service will disappear, and only metrics for the new current deployment will be shown.

For more information about container services, see [Container services in Amazon Lightsail](#).

Contents

- [Prerequisites](#)
- [Deployment parameters](#)
 - [Container entry parameters](#)
 - [Public endpoint parameters](#)
- [Communication between containers](#)
- [Container logs](#)
- [Deployment versions](#)
- [Deployment status](#)
- [Deployment failures](#)
- [View your current container service deployment](#)
- [Create or modify your container service deployment](#)

Prerequisites

Complete the following prerequisites before you get started with creating a deployment in your container service:

- Create your container service in your Lightsail account. For more information, see [Creating Amazon Lightsail container services](#).
- Identify the container images that you want to use when you launch containers on your container service.
 - Find container images on a public registry, such as the Amazon ECR Public Gallery. For more information, see [Amazon ECR Public Gallery](#) in the *Amazon ECR Public User Guide*.
 - Create container images on your local machine, then push them to your Lightsail container service. For more information, see the following guides:
 - [Installing software to manage container images for your Amazon Lightsail container services](#)
 - [Create container service images](#)
 - [Push and manage container images](#)

Deployment parameters

This section describes the parameters that you can specify for the container entries and the public endpoint of your deployment.

Container entry parameters

You can add up to 10 container entries in your deployment. Each container entry has the following parameters that you can specify:

Container name
Container names must contain only alphanumeric characters and hyphens. A hyphen (-) can separate words but cannot be at the start or end of the name.

Image
Enter the image reference from a public registry, such as DockerHub.

Configuration
Optionally specify a command, the environment variables, and the ports to open on your container.

Launch command:

Environment variables

Key	Value (optional)

✕

[+ Add variable](#)

Open ports
Your application code for this container must listen to a port specified here.

Port	Protocol
	HTTP ▼

✕

[+ Add port](#)

- **Container name** – Enter a name for the container. All containers within a deployment must have unique names, and must contain only alphanumeric characters and hyphens. A hyphen can separate words but it cannot be at the start or end of the name.
- **Source image** – Specify a source container image for the container. You can specify container images from the following sources:
 - A public registry, such as the Amazon ECR Public Gallery, or some other public container image registry.

For more information about Amazon ECR Public, see [What Is Amazon Elastic Container Registry Public?](#) in the *Amazon ECR Public User Guide*.

- Images pushed from your local machine to your container service. To specify a stored image, choose **Choose stored image**, and then select the image that you want to use.

If you create container images on your local machine, you can push them to your container service to use them when creating a deployment. For more information, see [Creating container images for your Amazon Lightsail container services](#) and [Pushing and managing container images on your Amazon Lightsail container services](#).

- **Launch command** – Specify a launch command to run a shell script or a bash script that configures your container when it's created. A launch command can do things like add software, update software, or configure your container in some other way.
- **Environment variables** – Specify environment variables, which are key-value parameters that provide dynamic configuration of the application or script run by the container.
- **Open ports** – Specify the ports and protocols to open on the container. You can specify to open any port over HTTP, HTTPS, TCP, and UDP. You must open an HTTP or HTTPS port for the container that you plan to use as the public endpoint of your container service. See the following section of this guide for more information.

Public endpoint parameters

You can specify the container entry in the deployment that will serve as the public endpoint of your container service. The application on the public endpoint container is publicly accessible on the internet through a randomly generated default domain of your container service. The default domain is formatted as `https://<ServiceName>.<RandomGUID>.<AWSRegion>.cs.amazonlightsail.com`, in which `<ServiceName>` is the name of your container service, `<RandomGUID>` is a randomly generated globally unique identifier of your container service in the AWS Region for your Lightsail account, and `<AWSRegion>` is the AWS Region in which the container service was created. The public endpoint of Lightsail container services supports HTTPS only, and it does not support TCP or UDP traffic. Only one container can be the public endpoint for a service. So make sure that you choose the container that is hosting the front-end of your application as the public endpoint while rest of the containers are internally accessible.

Note

You can use your own custom domain name with your container service. For more information, see [Enabling and managing custom domains for your Amazon Lightsail container services](#).

The public endpoint of your deployment, and container service, has the following parameters that you can specify:

The screenshot shows a configuration form titled "PUBLIC ENDPOINT". It includes the following elements:

- Instructions:** "Choose a container in your deployment that you want to make available to the internet as a public endpoint. Make sure to open an HTTP or HTTPS port on the selected container configuration, and then choose it as the port of your public endpoint."
- Warning:** An information icon followed by the text: "The container you choose as your public endpoint must respond to traffic on the specified port."
- Container Selection:** A dropdown menu with "nginx" selected.
- Port Selection:** A dropdown menu with "80" selected, accompanied by a question mark icon.
- Health check path:** A text input field containing the character "/".

- **Endpoint container** – Select the name of the container in your deployment that will serve as the public endpoint of your container service. Only the containers that have an HTTP or HTTPS port open in the deployment are listed in the dropdown menu.
- **Port** – Select the HTTP or HTTPS port to use for the public endpoint. Only the HTTP and HTTPS ports that are open on the selected container are listed in the dropdown menu. Select an HTTP port if the selected container is not configured to support an HTTPS connection when first launched.

Note

The default domain for your container service uses HTTPS by default even if you choose an HTTP port as the public endpoint port. This is because the load balancer of your container service is configured for HTTPS by default, but it uses HTTP to establish a connection with your containers.

The load balancer of your container service connects to your containers using HTTP, but serves content to users using HTTPS.

- **Health check path** – Specify a path on the selected public endpoint container where your container service's load balancer will periodically check to make sure it's healthy.
- **Advanced health check settings** – You can configure the following health check settings for the selected public endpoint container:
 - **Health check timeout seconds** - The amount of time, in seconds, to wait for a response. If no response is received during this time, the health check fails. You can specify 2–60 seconds.
 - **Health check interval seconds** - The approximate interval, in seconds, between health checks of the container. You can specify 5–300 seconds.
 - **Health check success codes** - The HTTP codes to use when checking for a successful response from a container. You can specify values between 200 and 499. You can specify multiple values (for example, 200,202) or a range of values (for example, 200–299).
 - **Health check healthy threshold** - The number of consecutive health check successes required before moving the container to the Healthy state.
 - **Health check unhealthy threshold** - The number of consecutive health check failures required before moving the container to the Unhealthy state.

Private domain

All container services also have a private domain that is formatted as `<ServiceName>.service.local`, in which `<ServiceName>` is the name of your container service. Use the private domain to access your container service from another one of your Lightsail resources in the same AWS Region as your service. The private domain is the only way to access your container service if you don't specify a public endpoint in the deployment of your service. A default domain is generated for your container service even if you don't specify a public endpoint, but it will show a 404 No Such Service error message when you try to browse to it.

To access a specific container using the private domain of your container service, you must specify the open port of the container that will accept your connection request. You do this by formatting the domain of your request as `<ServiceName>.service.local:<PortNumber>`, in which `<ServiceName>` is the name of your container service and `<PortNumber>` is the open port of the container that you wish to connect to. For example, if you create a deployment on your container service named `container-service-1`, and you specify a Redis container with port 6379 open, then you should format the domain of your request as `container-service-1.service.local:6379`.

Communication between containers

Using environment variables, you can open communications between containers within the same container service, containers within different container services, or between a container and other resources (for example, between a container and a managed database).

To open communication between containers within the same container service, add an environment variable to your container deployment that references `localhost` as shown in the following example.

Key	Value (optional)
SERVICE_CON	service://localhost

To open communication between containers that are in different container services, add an environment variable to your container deployment that references the private domain (for example, `container-service-1.service.local`) of the other container service as shown in the following example.

Key	Value (optional)
SERVICE_CON	service://container-service-1.service.local

To open communication between containers and other resources, add an environment variable to your container deployment that references the public endpoint URL of the resource. For example, the public endpoint of a Lightsail managed database is typically `ls-123abc.czoexamplezqi.us-west-2.rds.amazonaws.com`. So you should reference that in the environment variable as shown in the following example.

Key	Value (optional)
WORDPRESS_	ls-123abc.czoexamplezqi.us-west-2.rds.amazon

Container logs

Every container in your deployment generates a log. The container logs provide the *stdout* and *stderr* streams of processes that run inside the container. Access your containers' logs periodically

to diagnose their operations. For more information, see [Viewing the container logs of your Amazon Lightsail container services](#).

Deployment versions

Every deployment that you create in your container service is saved as a deployment version. If you modify the parameters of an existing deployment, the containers are re-deployed to your service and the modified deployment results in a new deployment version. The latest 50 deployment versions for each container service are saved. You can use any of the 50 deployment versions to create a new deployment in the same container service. For more information, see [Viewing and managing deployment versions of your Amazon Lightsail container services](#).

Deployment status

Your deployment can be in one of the following states after it's created:

- **Activating** – Your deployment is activating and your containers are being created.
- **Active** – Your deployment was successfully created, and it's currently running on your container service.
- **Inactive** – Your previously successfully created deployment is no longer running on your container.
- **Failed** – Your deployment failed because one or more of the containers specified in the deployment failed to launch.

Deployment failures

Your deployment fails if one or more containers in your deployment fails to launch. If your deployment fails, and there is a previous deployment running on your container service, then your container service keeps the previous deployment as the active deployment. If there is no previous deployment, then your container service remains in ready state with no currently active deployment.

View the container logs of the failed deployment to diagnose and troubleshoot what went wrong. For more information, see [Viewing the container logs of your Amazon Lightsail container services](#).

View your current container service deployment

Complete the following procedure to view the current deployment on your Lightsail container service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to view the current deployment.
4. On the container service management page, choose the **Deployments** tab.

The **Deployments** page lists your current deployment and deployment versions. Both sections of the page are empty if you haven't created a deployment in your container service.

Create or modify your container service deployment


Complete the following procedure to create or modify a deployment on your Lightsail container service. Whether you create a new deployment or modify an existing one, your container service saves your every deployment as a new deployment version. For more information, see [Viewing and managing deployment versions of your Amazon Lightsail container services](#).


1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to create or modify a container service deployment.
4. On the container service management page, choose the Deployments tab.

The **Deployments** page lists your current deployment and deployment versions, if any.


5. Choose one of the following options:
 - If your container service has an existing deployment, choose **Modify your deployment**.
 - If your container service has not had a deployment, choose **Create a deployment**.

The deployment form opens, where you can edit existing deployment parameters, or enter new deployment parameters.

 **Create your first deployment**

 Saving this deployment will create a new deployment version

CONTAINERS

Remove 

Container name
Container names must contain only alphanumeric characters and hyphens. A hyphen (-) can separate words but cannot be at the start or end of the name.

Image
Enter the image reference from a public registry, such as DockerHub.


Configuration
Optionally specify a command, the environment variables, and the ports to open on your container.

Launch command:


[+ Add environment variables](#)


[+ Add open ports](#)



[+ Add container entry](#)

 You can have up to 10 containers in a deployment

PUBLIC ENDPOINT
You must specify container names for the container entries in your deployment to be able to select a container as the public endpoint of your deployment.

 The container you choose as your public endpoint must respond to traffic on the specified port.



[Cancel](#)  [Save and deploy](#) 

6. Enter the parameters of your deployment. For more information about the deployment parameters that you can specify, see the [Deployment parameters](#) section earlier in this guide.
7. Choose **Add container entry** to add more than one container entry to your deployment. You can have up to 10 container entries in your deployment.
8. Choose the container entry of your deployment to serve as the public endpoint container service. This includes specifying the HTTP or HTTPS port, the health check path on the selected container entry, and advanced health check settings. For more information, see [Public endpoint parameters](#) earlier in this guide.

9. When you're done entering the parameters of your deployment, choose **Save and deploy** to create the deployment on your container service.

The status of your container service changes to **Deploying** while your deployment is being created. After a few moments, the status of your container service changes to one of the following depending on the status of your deployment:

- If your deployment succeeds, the status of your container service changes to **Running** and the status of the deployment changes to **Active**. If you configured a public endpoint in your deployment, then the container chosen as the public endpoint is available through the default domain of your container service.
- If your deployment fails, and there is a previous deployment running on your container service, the status of your container service changes to **Running** and your container service keeps the previous deployment as the active deployment. If there is no previous deployment, the status of your container service changes to **Ready** with no currently active deployment. View the container logs of the failed deployment to diagnose and troubleshoot what went wrong. For more information, see [Viewing the container logs of your Amazon Lightsail container services](#).

Topics

- [Change the capacity of your Lightsail container service](#)
- [Manage deployment versions of a Lightsail container service](#)
- [View Lightsail container service logs](#)

Change the capacity of your Lightsail container service

The capacity of your Amazon Lightsail container service is made up of its scale and power. The scale specifies the number of compute nodes in your container service, and the power specifies the memory and vCPUs of each node in your service. You pick the scale based on the number of nodes you want powering your service for better availability and higher capacity

By following the procedure in this guide, you can dynamically increase the power and scale of your container service at any time without any down-time if you find that it's under-provisioned, or decrease it if you find that it's over-provisioned. Lightsail automatically manages the capacity change along with your current deployment.

Note

If you create a new deployment, then the existing utilization metrics of your container service will disappear, and only metrics for the new current deployment will be shown.

For more information about container services, see [Container services](#).

Change the capacity of your container service

Complete the following procedure to change the capacity of your Lightsail container service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to change the capacity.
4. On the container service management page, choose the **Capacity** tab.

The current power, scale, and monthly price of your container service is displayed in the **Capacity** page.

5. Choose **Change capacity** to change the power and scale to something else.
6. On the confirmation prompt that appears, choose **Yes, continue** to acknowledge that changing the capacity of your container service will re-deploy the current deployment.
7. Choose the new power and scale of your container service.
8. Choose **Yes, apply** to apply the new capacity to your container service.

The status of your container service changes to **Updating**. After a few moments, the status of your service changes to **Enabled**, and it begins operating under its new capacity.

Manage deployment versions of a Lightsail container service

Every deployment that you create in your Amazon Lightsail container service is saved as a deployment version. If you modify the parameters of an existing deployment, the containers are re-deployed to your service and the modified deployment results in a new deployment version. The latest 50 deployment versions for each container service are saved. You can use any of the 50 deployment versions to create a new deployment in the same container service. In this guide, we show you how to view and manage the deployment versions of your container service.

For more information about container services, see [Container services](#).

Deployment version status

Each of your deployment versions can be in one of the following states after it's created:

- **Deploying (Activating)** – The deployment is being launched.
- **Active** – Your deployment was successfully created, and it's currently running on your container service. Your container service can have only one deployment in an active state at a time.
- **Inactive** – Your previously successfully created deployment is no longer running on your container.
- **Failed** – Your deployment failed because one or more of the containers specified in the deployment failed to launch.

Prerequisites

Before you get started, you need to create a Lightsail container service. For more information, see [Create a container service](#).

You also should create a deployment in your container service that configures and launches your containers. For more information, see [Creating and managing deployments for your Amazon Lightsail container services](#).

View the deployment versions of a container service

Complete the following procedure to view the deployment versions of your Lightsail container service.


1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to view the deployment versions.
4. On the container service management page, choose the **Deployments** tab.

The **Deployments** page lists your current deployment and deployment versions, if any.

5. The deployment versions of your container service are listed under the **Deployment versions** section of the page.

Each deployment has a date, in which it was created, a status, and an actions menu.

- Choose one of the following options through the actions menu of a deployment version:
 - Create new deployment** – Choose this option to create a new deployment from the selected deployment version. For more information about creating a deployment, see [Create or modify your container service deployment](#).

 **Note**

If you choose to create a new deployment from a version that has a **Failed** status, then you must correct the cause of the failure before creating the deployment. Otherwise, the deployment will likely fail again.

- View details** – Choose this option to view the container entry and public endpoint parameters of the selected deployment version. You can also view the container logs for the deployment in case you need to diagnose a failed deployment. For more information, see [View container service logs](#).

View Lightsail container service logs

Every container in your Amazon Lightsail container service deployment generates a log. The container logs provide the stdout and stderr streams of processes that run inside your containers. Access your containers' logs periodically to diagnose their operations. The latest three days of log entries are stored before the oldest ones are replaced by the newest entries.

Filter container logs

Container logs can have hundreds of entries per day. Use the filtering options to reduce the number of entries displayed in your log window, and make it easier to find what you're looking for. You can filter container logs by a start and end date (in local time), and by a specific term. When filtering by a term, you can choose to include or exclude log entries for the term you specify.



The screenshot shows the 'Logs for nginx' interface. At the top right is a 'Refresh' button with a circular arrow icon. Below the title, there is a note: 'Log events are in Coordinated Universal Time (UTC)'. The main area contains a 'Display entries from' section with a calendar icon and a date range selector. To the right is a 'Filter' section with a text input field containing 'Example: [DEBUG]' and an 'Apply' button. Below the filter input are radio buttons for 'Include' (selected) and 'Exclude'. At the bottom left, there is a link: 'Learn more about container logs' with an external link icon.

The *include* or *exclude* filter term looks for an exact match that is case-sensitive. For example, if you specify to include only log events that have HTTP in the message, then you will see all log events that include HTTP in the message, but none that include `http` in the message. If you specify to exclude `ERROR`, then you will see all log events that don't include `ERROR` in the message, and you will also see log events that include `ERROR` in the message.

Prerequisites

Before you get started, you need to create a Lightsail container service. For more information, see [Creating Amazon Lightsail container services](#).

You also should create a deployment in your container service that configures and launches your containers. For more information, see [Creating and managing deployments for your Amazon Lightsail container services](#).

View the container logs

Complete the following procedure to view the container logs of your Lightsail container service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to view the container logs.
4. On the container service management page, choose the **Deployments** tab.

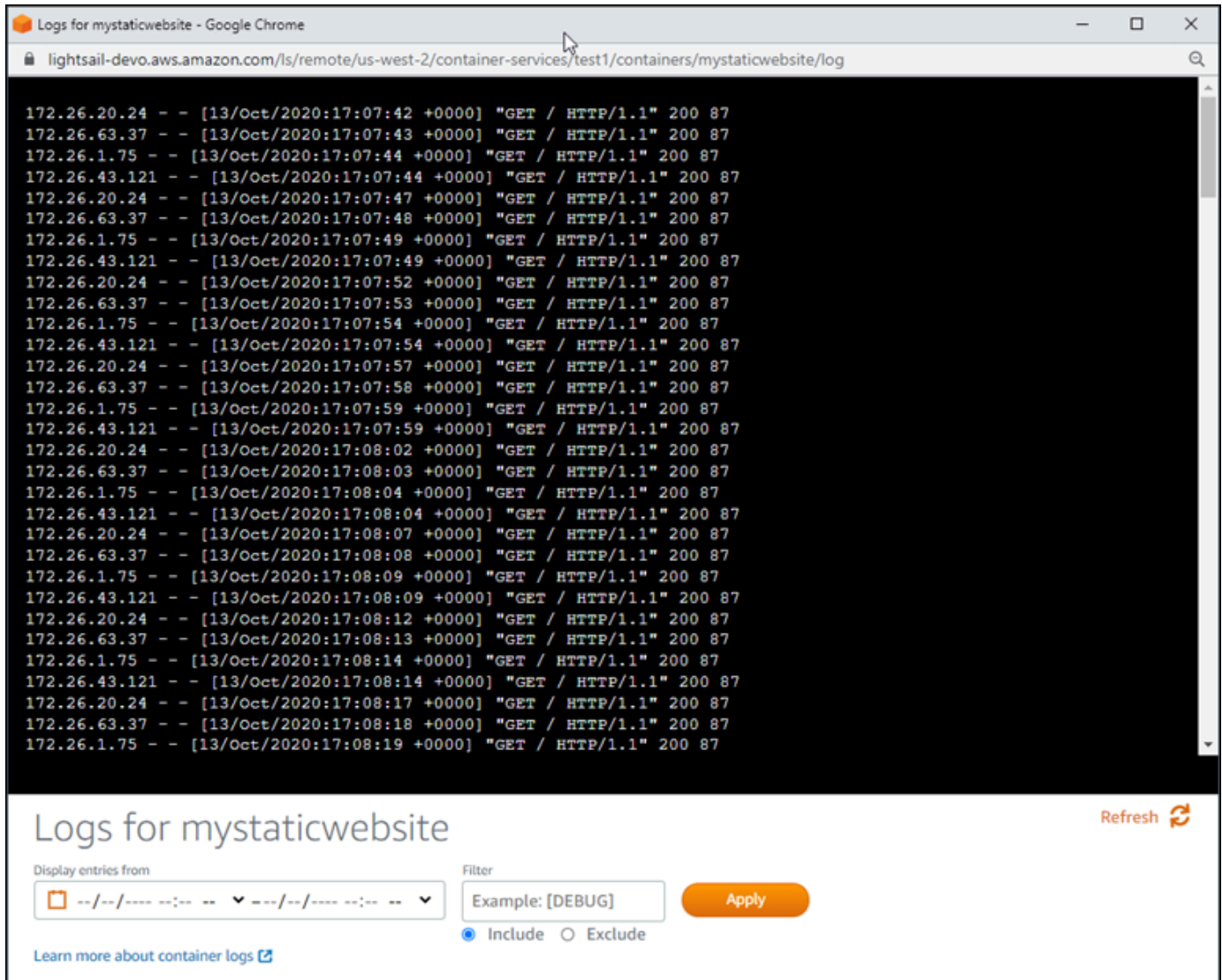
The **Deployments** page lists your current deployment and deployment versions, if any.

5. Choose one of the following options to view container logs:
 - To access the container logs of the current deployment, choose **Open log** for the container entries under the **Current deployment** section of the page.
 - To access the container logs of a previous deployment, choose the actions menu icon (:) for a previous deployment under the **Deployment versions** section of the page, and then choose **Show details**. In the **Version details** page that appears, choose **Open log** for the container entries that are listed.

The container log opens in a new browser window. You can scroll down to view more log entries, and refresh the page to load the newest set of entries. The filtering options are displayed at the bottom of the page.

Note

Log entries are displayed in ascending order, and in Coordinated Universal Time (UTC). That is, the oldest log entries are at the top, and you must scroll down to see newer log entries.



The screenshot shows a Google Chrome browser window with the URL `lightsail-dev0.aws.amazon.com/ls/remote/us-west-2/container-services/test1/containers/mystaticwebsite/log`. The main content area displays a list of log entries for the 'mystaticwebsite' container. Each entry follows the format: `IP - - [timestamp] "method / HTTP/1.1" status code`. The logs show a series of 'GET' requests from various IP addresses (e.g., 172.26.20.24, 172.26.63.37, 172.26.1.75) at regular intervals, all resulting in a status code of 200. Below the log list, there is a control panel with the title 'Logs for mystaticwebsite' and a 'Refresh' button. The control panel includes a 'Display entries from' dropdown menu, a 'Filter' input field containing 'Example: [DEBUG]', and an 'Apply' button. Below the filter, there are radio buttons for 'Include' (selected) and 'Exclude'. A link 'Learn more about container logs' is also present.

Enable and manage custom domains in Lightsail

Enable custom domains for your Amazon Lightsail container service to use your registered domain names with your service. Before you enable custom domains, your container service accepts traffic

only for the default domain that is associated with your service when you first create it (e.g., `containerservicename.123456abcdef.us-west-2.cs.amazonlightsail.com`). When you enable custom domains, you choose the Lightsail SSL/TLS certificate that you created for the domains that you want to use with your container service, and then you choose the domains you want to use from that certificate. After you enable custom domains, your container service accepts traffic for all of the domains that are associated with the certificate that you chose.

Important

If you choose a Lightsail container service as the origin of your distribution, Lightsail automatically adds the default domain name of your distribution as a custom domain on your container service. This enables traffic to be routed between your distribution and your container service. However, there are some circumstances in which you might need to manually add the default domain name of your distribution to your container service. For more information, see [Add the default domain of a distribution to a container service](#).

Contents

- [Container service custom domain limits](#)
- [Prerequisites](#)
- [View custom domains for a container service](#)
- [Enable custom domains for a container service](#)
- [Disable custom domains for a container service](#)

Container service custom domain limits

The following limits apply to container service custom domains:

- You can use up to 4 custom domains with each of your Lightsail container services, and you cannot use the same domains on more than one service.
- If you use a Lightsail DNS zone to manage the DNS of your domain, then you can route traffic for the apex of your domain (e.g., `example.com`) and for subdomains (e.g., `www.example.com`) to your container services.

Prerequisites

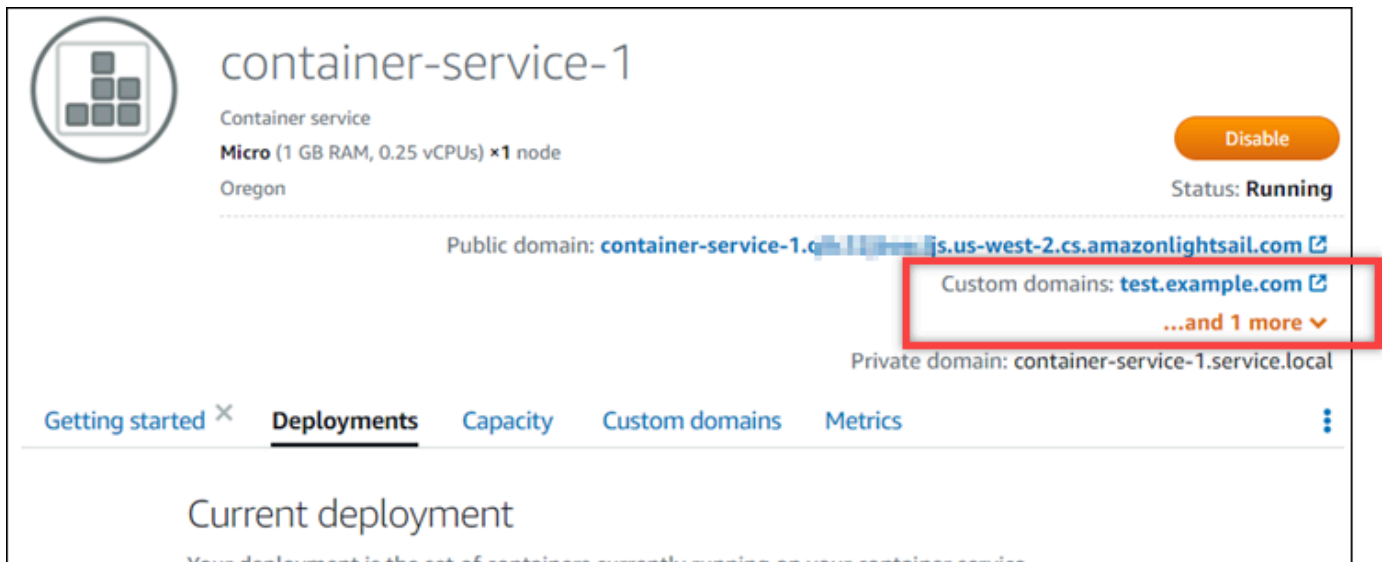
Before you get started, you need to create a Lightsail container service. For more information, see [Creating Amazon Lightsail container services](#).

You also should have created and validated an SSL/TLS certificate for your container service. For more information, see [Create container service SSL/TLS certificates](#) and [Validate container service SSL/TLS certificates](#).

View custom domains for a container service

Complete the following procedure to view the custom domains that are currently enabled for your container service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to view the enabled custom domains.
4. Locate the custom domain values in the heading of the container service management page, as shown in the following example. These are the custom domains that are currently enabled for the container service.



5. On the container service management page, choose the **Custom domains** tab.

The custom domains being used under each attached certificate, are listed under the **Custom domain SSL/TLS certificates** section of the page. The certificates currently attached to your container service, are listed under the **Attached certificates** section.

Enable custom domains for a container service

Complete the following procedure to enable custom domains for your Lightsail container service by attaching a certificate to your service.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to enable custom domains.
4. On the container service management page, choose the **Custom domains** tab.

The **Custom domains** page displays the SSL/TLS certificates currently attached to your container service, if any.

5. Choose **Attach certificate**.

If you have no certificates, then you must first create and validate an SSL/TLS certificate for your domains, before you can attach it to your container service. For more information, see [Create container service SSL/TLS certificates](#).

6. In the dropdown menu that appears, select a valid certificate for the domain(s) that you want to use with your container service.
7. Verify the certificate information is correct, then choose **Attach**.
8. The container service's **Status** will change to **Updating**. After the status changes to **Ready**, the certificate's domain will appear in the **Custom domains** section.
9. Choose **Add domain assignment** to point the domain to your container service.
10. Verify the certificate and DNS information are correct, then choose **Add assignment**. After a few moments, traffic for the domain that you selected will begin to be accepted by your container service.
11. After you've added the domain assignment, open a new browser window and browse to the custom domain that you enabled for your container service. The application that is running on your container service, if any, should load.

Disable custom domains for a container service

Complete the following procedure to disable custom domains for your Lightsail container service by detaching a certificate from your service, or by deselecting a previously selected domain.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container service for which you want to disable custom domains.
4. On the container service management page, choose the **Custom domains** tab.

The **Custom domains** page displays the SSL/TLS certificates currently attached to your container service, if any.

5. Choose one of the following options:
 1. Choose **Configure container service domains** to either deselect domains that were previously selected, or to select more domains that are associated to the container service.
 2. Choose **Detach** to detach the certificate from the container service, and remove all of its associated domains from the service.

Important

If you haven't already done so, modify the DNS records of your domain so that traffic routes stops routing to your container service and instead routes to another resource.

Topics

- [Route traffic for your domain to a Lightsail container service](#)
- [Route traffic for a domain in Route 53 to a Lightsail container service](#)

Route traffic for your domain to a Lightsail container service

You must point your registered domain names to your Amazon Lightsail container service after you enabled custom domains for your service. You do this by adding an alias record to the DNS zone of each of the domains specified on the certificates that you're using with your container service. All of the records that you add should point to the default domain (e.g., `https://`

<ServiceName>.<RandomGUID>.<AWSRegion>.cs.amazonlightsail.com) of your container service.

In this guide, we provide you with the procedure to point your domains to your container service using a Lightsail DNS zone. For more information about Lightsail DNS zones, see [DNS in Amazon Lightsail](#).

For more information about container services, see [Container services](#).

Note

If you're using Route 53 to host the DNS of your domain, then you should add the alias record to the hosted zone of your domain in Route 53. For more information, see [Routing traffic for a domain in Route 53 to an Amazon Lightsail container service](#).

Prerequisite

Before you get started, you should enable custom domains for your Lightsail container service. For more information, see [Enabling and managing custom domains for your Amazon Lightsail container services](#).

Get the default domain of your container service

Complete the following procedure to get default domain name of your container service, which you specify when you add an alias record to the DNS of your domain.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of a container service for which want get the default domain name.
4. In the header section of your container service management page, make note of your default domain name. Your container service default domain name is similar to `<ServiceName>.<RandomGUID>.<AWSRegion>.cs.amazonlightsail.com`.

You must add this value as part of a canonical name (CNAME) record in the DNS of your domains. We recommend that you copy and paste this value into a text file that you can refer to later. For more information, see the following [Add the CNAME records to your domain's DNS zone](#) section of this guide.

Add a record to your domain's DNS zone

Complete the following procedure to add an address (A for IPv4 or AAAA for IPv6) record, or canonical (CNAME) record to your domain's DNS zone.

1. On the Lightsail home page, choose the **Domains & DNS** tab.
2. Under the **DNS zones** section of the page, choose the domain name to which you want to add the record that will direct traffic for your domain to your container service.
3. Choose the **DNS records** tab.
4. Complete one of the following steps depending on the current state of your DNS zone:
 - If you haven't added an A, AAAA, or CNAME record, choose **Add record**.
 - If you previously added an A, AAAA, or CNAME record, choose the edit icon next to the existing A, AAAA, or CNAME record listed on the page, and then skip to step 5 of this procedure.
5. Choose **A record**, **AAAA record**, or **CNAME record** in the **Record type** dropdown menu.
 - Add an A record to map the apex of your domain (e.g., `example.com`) or a subdomain (e.g., `www.example.com`) to your container service under the IPv4 network.
 - Add an AAAA record to map the apex of your domain (e.g., `example.com`) or a subdomain (e.g., `www.example.com`) to your container service under the IPv6 network.
 - Add a CNAME record to map a subdomain (e.g., `www.example.com`) to the public domain (default DNS) of your container service.
6. In the **Record name** text box, enter one of the following options:
 - For an A record or AAAA record, enter `@` to route traffic for the apex of your domain (e.g., `example.com`) to your container service, or enter a subdomain (e.g., `www`) to route traffic for a subdomain (e.g., `www.example.com`) to your container service.
 - For a CNAME record, enter a subdomain (e.g., `www`) to route traffic for a subdomain (e.g., `www.example.com`) to your container service.
7. Complete one of the following steps depending on the record you're adding:
 - For an A record or AAAA record, choose the name of your container service in the **Resolves to** text box.
 - For a CNAME record, enter the default domain name of your container service into the **Maps to** text box.

8. Choose the save icon to save the record to your DNS zone.

Repeat these steps to add additional DNS records for domains on your certificate that you are using with your container service. Allow time for changes to propagate through the Internet's DNS. After a few minutes, you should see if your domain is pointing to your container service.

Route traffic for a domain in Route 53 to a Lightsail container service

You can route traffic for a registered domain, such as `example.com`, to the applications running on a Lightsail container service. You do this by adding an alias record to the hosted zone of your domain that points to the default domain of your Lightsail container service.

In this tutorial, we show you how to add an alias record for your Lightsail container service to a hosted zone in Route 53. You can do this only by using the AWS Command Line Interface (AWS CLI). It cannot be done using the Route 53 console.

Note

If you're using Lightsail to host the DNS of your domain, then you should add the alias record to the DNS zone of your domain in Lightsail. For more information, see [Routing traffic for a domain in Amazon Lightsail to a Lightsail container service](#).

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Get the hosted zone IDs for Lightsail container services](#)
- [Step 3: Create a record set JSON file](#)
- [Step 4: Add a record to the hosted zone of your domain in Route 53](#)

Step 1: Complete the prerequisites

Complete the following prerequisites if you haven't already:

- Register a domain name in Route 53, or make Route 53 the DNS service for your registered (existing) domain name. For more information, see [Registering domain names using Amazon Route 53](#) or [Making Amazon Route 53 the DNS service for an existing domain](#) in the *Amazon Route 53 Developer Guide*.

- Deploy your applications to your Lightsail container service. For more information, see [Create and manage container service deployments](#).
- Enable your registered domain name on your Lightsail container service. For more information, see [Enable and manage custom domains](#).
- Configure the AWS CLI with your account. For more information, see [Configure the AWS CLI to work with Lightsail](#).

Step 2: Get the hosted zone IDs for Lightsail container services

You must specify a hosted zone ID for your Lightsail container service when you add an alias record to a hosted zone in Route 53. For example, if your Lightsail container service is in the US West (Oregon) (us-west-2) AWS Region, then you must specify hosted zone ID Z0959753D43BBB908BAV when adding an alias record for your Lightsail container service to a hosted zone in Route 53.

Following are the hosted zone IDs for each AWS Region in which you can create a Lightsail container service.

EU (London) (eu-west-2): Z0624918ZXDYQZLOXA66

US East (N. Virginia) (us-east-1): Z06246771KYU0IRHI74W4

Asia Pacific (Singapore) (ap-southeast-1): Z0625921354DRJH4EY9V0

EU (Ireland) (eu-west-1): Z0624732FELAMMKW3Y21

Asia Pacific (Tokyo) (ap-northeast-1): Z0626125UAU4JWQ9JSKN

Asia Pacific (Seoul) (ap-northeast-2): Z06260262XZM84B2WPLHH

Asia Pacific (Mumbai) (ap-south-1): Z10460781IQMISS0I0VVY

Asia Pacific (Sydney) (ap-southeast-2): Z09597943PQQZATPFE96E

Canada (Central) (ca-central-1): Z10450993RIRIJJUUMA5W

Europe (Frankfurt) (eu-central-1): Z06137433FV04OY4EC6L0

Europe (Stockholm) (eu-north-1): Z016970523TDG2TZMUXKK

Europe (Paris) (eu-west-3): Z09594631DSW2QUR7CFGO

US East (Ohio) (us-east-2): Z10362273VJ548563IY84

US West (Oregon) (us-west-2): Z0959753D43BBB908BAV

Step 3: Create a record set JSON file

When you add a DNS record to the hosted zone of your domain in Route 53 using the AWS CLI, you must specify a set of configuration parameters for the record. The easiest way to do this is by creating a JSON (.json) file that contains all of the parameters, and then referencing the JSON file in your AWS CLI request.

Complete the following procedure to create a JSON file with the record set parameters for the alias record:

1. Open a text editor, such as Notepad on Windows or Nano on Linux.
2. Copy and paste the following text into the text editor:

```
{
  "Comment": "Comment",
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "Domain.",
        "Type": "A",
        "AliasTarget": {
          "HostedZoneId": "LightsailContainerServiceHostedZoneID",
          "DNSName": "LightsailContainerServiceAddress.",
          "EvaluateTargetHealth": true
        }
      }
    }
  ]
}
```

In your file, replace the following example text with your own:

- *Comment* with a personal note or comment about the record set.
- *Domain* with the registered domain name that you want to use with your Lightsail container service (for example, `example.com` or `www.example.com`). To use the root of your domain with your Lightsail container service, you must specify an `@` symbol in the subdomain space of your domain (for example, `@.example.com`).

- `LightsailContainerServiceHostedZoneID` with the hosted zone ID for the AWS Region in which you created your Lightsail container service. For more information, see [Step 2: Get the hosted zone IDs for Lightsail container services](#) earlier in this guide.
- `LightsailContainerServiceAddress` with the public domain name of your Lightsail container service. You can get this by signing in to the Lightsail console, browsing to your container service, and copying the **Public domain** listed in the header section of the container service management page (for example, `container-service-1.q8cexampleljs.us-west-2.cs.amazonlightsail.com`).

Example:

```
{
  "Comment": "Alias record for Lightsail container service",
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "@.example.com.",
        "Type": "A",
        "AliasTarget": {
          "HostedZoneId": "Z0959753D43BBB908BAV",
          "DNSName": "container-service-1.q8cexampleljs.us-
west-2.cs.amazonlightsail.com.",
          "EvaluateTargetHealth": true
        }
      }
    }
  ]
}
```

3. Save the file to your local directory as `change-resource-record-sets.json`.

Step 4: Add a record to the hosted zone of your domain in Route 53

Complete the following procedure to add a record to the hosted zone of your domain in Route 53 using the AWS CLI. You do this by using the `change-resource-record-sets` command. For more information, see [change-resource-record-sets](#) in the *AWS CLI Command Reference*.

Note

You must install the AWS CLI and configure it for Lightsail and Route 53 before continuing with this procedure. For more information, see [Configure the AWS CLI to work with Lightsail](#).

1. Open a Command Prompt or Terminal window.
2. Enter the following command to add a record to the hosted zone of your domain in Route 53.

```
aws route53 change-resource-record-sets --hosted-zone-id HostedZoneID --change-batch PathToJsonFile
```

In the command, replace the following example text with your own:

- *HostedZoneID* with the ID of the hosted zone for your registered domain in Route 53. Use the [list-hosted-zones](#) command to get a list of IDs for the hosted zones in your Route 53 account.
- *PathToJsonFile* with the local directory folder path on your computer of the .json file that contains the record parameters. For more information, see the [Step 3: Create a record set JSON file](#) section earlier in this guide.

Examples:

On a Linux or Unix computer:

```
aws route53 change-resource-record-sets --hosted-zone-id Z123456789ABCDEFGHIJ --change-batch home/user/awscli/route53/change-resource-record-sets.json
```

On a Windows computer:

```
aws route53 change-resource-record-sets --hosted-zone-id Z123456789ABCDEFGHIJ --change-batch file:///C:\awscli\route53\change-resource-record-sets.json
```

You should see a result similar to the following example:

```
H:\>aws route53 change-resource-record-sets --hosted-zone-id Z123456789ABCDEFGHIJ
--change-batch file://C:\awscli\route53\change-resource-record-sets.json
-
{
  "ChangeInfo": {
    "Id": "/change/C05953EXAMPLEZ4V4LOAC",
    "Status": "PENDING",
    "SubmittedAt": "2021-08-11T20:58:30.960000+00:00",
    "Comment": "Alias record for Lightsail container service"
  }
}
```

Allow time for the change to propagate through the internet's DNS, which might take several hours. After that is completed, internet traffic for your registered domain in Route 53 should begin routing to your Lightsail container service.

Security in Amazon Lightsail

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. To learn about the compliance programs, and which services they apply to, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Amazon Lightsail. The following topics show you how to configure Amazon Lightsail to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon Lightsail resources.

Infrastructure security in Amazon Lightsail

As a managed service, Amazon Lightsail is protected by the AWS global network security. For information about AWS security services and how AWS protects infrastructure, see [AWS Cloud Security](#). To design your AWS environment using the best practices for infrastructure security, see [Infrastructure Protection](#) in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Lightsail through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

Resilience in Amazon Lightsail

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see [AWS Global Infrastructure](#).

In addition to the AWS global infrastructure, Amazon Lightsail offers several features to help support your data resiliency and backup needs.

- Copying instance and disk snapshots across Regions. For more information, see [Snapshots](#).
- Automating instance and disk snapshots. For more information, see [Snapshots](#).
- Distributing incoming traffic across multiple instances in a single Availability Zone or multiple Availability Zones using a load balancer. For more information, see [Load balancers](#).

Identity and access management for Amazon Lightsail

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work you do in Amazon Lightsail.

Service user – If you use the Amazon Lightsail service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon Lightsail features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Amazon Lightsail, see [Troubleshoot Identity and Access Management \(IAM\)](#).

Service administrator – If you're in charge of Amazon Lightsail resources at your company, you probably have full access to Amazon Lightsail. It's your job to determine which Amazon Lightsail features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Amazon Lightsail, see [How Amazon Lightsail Works with IAM](#).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Amazon Lightsail. To view example Amazon Lightsail identity-based policies that you can use in IAM, see [Amazon Lightsail Identity-Based Policy Examples](#).

Authenticating With Identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see [The IAM Console and Sign-in Page](#) in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication, or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the [AWS Management Console](#), use your password with your root user email or your IAM user name. You can access AWS programmatically using your root user or IAM user access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see [Signature Version 4 Signing Process](#) in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see [Using Multi-Factor Authentication \(MFA\) in AWS](#) in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see [Tasks that require root user credentials](#) in the *IAM User Guide*.

IAM Users and Groups

An [IAM user](#) is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see [Rotate access keys regularly for use cases that require long-term credentials](#) in the *IAM User Guide*.

An [IAM group](#) is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see [When to create an IAM user \(instead of a role\)](#) in the *IAM User Guide*.

IAM Roles

An [IAM role](#) is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by [switching roles](#). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see [Using IAM roles](#) in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see [Creating a role for a third-party Identity Provider](#) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permission sets, see [Permission sets](#) in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
 - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see [Forward access sessions](#).
 - **Service role** – A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.
 - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked

roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see [When to create an IAM role \(instead of a user\)](#) in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see [Creating a role for a third-party Identity Provider](#) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permission sets, see [Permission sets](#) in the *AWS IAM Identity Center User Guide*.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.

- **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see [Actions, Resources, and Condition Keys for Amazon Lightsail](#) in the *Service Authorization Reference*.
- **Service role** – A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.
- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see [When to create an IAM role \(instead of a user\)](#) in the *IAM User Guide*.

Managing Access Using Policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see [Overview of JSON policies](#) in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-Based Policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Creating IAM policies](#) in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see [Choosing between managed policies and inline policies](#) in the *IAM User Guide*.

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Creating IAM policies](#) in the *IAM User Guide*.

Resource-Based Policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are *IAM role trust policies* and *Amazon S3 bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must [specify a principal](#) in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are *IAM role trust policies* and *Amazon S3 bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must [specify a principal](#) in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Access Control Lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see [Access control list \(ACL\) overview](#) in the *Amazon Simple Storage Service Developer Guide*.

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Other Policy Types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs work](#) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see [Session policies](#) in the *IAM User Guide*.
- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to

any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs work](#) in the *AWS Organizations User Guide*.

- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see [Session policies](#) in the *IAM User Guide*.

Multiple Policy Types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see [Policy evaluation logic](#) in the *IAM User Guide*.

Topics

- [AWS managed policies for Amazon Lightsail](#)
- [How Amazon Lightsail works with IAM](#)
- [Manage access to Amazon Lightsail for an IAM user](#)

AWS managed policies for Amazon Lightsail

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to [create IAM customer managed policies](#) that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see [AWS managed policies](#) in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see [AWS managed policies for job functions](#) in the *IAM User Guide*.

AWS managed policy: LightsailExportAccess

You can't attach LightsailExportAccess to your IAM entities. This policy is attached to a service-linked role that allows Lightsail to perform actions on your behalf. For more information, see [Service-linked roles](#).

This policy grants permissions that allow Lightsail to export your instance and disk snapshots to Amazon Elastic Compute Cloud, and get the current account-level Block Public Access configuration from Amazon Simple Storage Service (Amazon S3).

Permissions details

This policy includes the following permissions.

- **ec2** – Allows access to list and copy instance images and disk snapshots.
- **iam** – Allows access to delete service-linked roles and retrieve the status of your service-linked role deletion.
- **s3** – Allows access to retrieve the PublicAccessBlock configuration for an AWS account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
      ],
      "Resource": "arn:aws:iam::*:role/aws-service-role/lightsail.amazonaws.com/AWSServiceRoleForLightsail*"
    },
    {
      "Effect": "Allow",
      "Action": [
```



```
"ec2:CopySnapshot",
"ec2:DescribeSnapshots",
"ec2:CopyImage",
"ec2:DescribeImages"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": [
"s3:GetAccountPublicAccessBlock"
],
"Resource": "*"
}
]
```

Lightsail updates to AWS managed policies

- Edit to the LightsailExportAccess managed policy

Added the `s3:GetAccountPublicAccessBlock` action to the `LightsailExportAccess` managed policy. It allows Lightsail to get the current account-level Block Public Access configuration from Amazon S3.

January 14, 2022

- Lightsail started tracking changes

Lightsail started tracking changes for its AWS managed policies.

January 14, 2022

How Amazon Lightsail works with IAM

Before you use IAM to manage access to Lightsail, you should understand what IAM features are available to use with Lightsail. To get a high-level view of how Lightsail and other AWS services work with IAM, see [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Lightsail Identity-Based Policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. Lightsail supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see [IAM JSON Policy Elements Reference](#) in the *IAM User Guide*.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Action element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in Lightsail use the following prefix before the action: `lightsail:`. For example, to grant someone permission to run a Lightsail instance with the Lightsail CreateInstances API operation, you include the `lightsail:CreateInstances` action in their policy. Policy statements must include either an Action or NotAction element. Lightsail defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [  
    "lightsail:action1",  
    "lightsail:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word Create, include the following action:

```
"Action": "lightsail:Create*"
```

To see a list of Lightsail actions, see [Actions Defined by Amazon Lightsail](#) in the *IAM User Guide*.

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Resource JSON policy element specifies the object or objects to which the action applies. Statements must include either a Resource or a NotResource element. As a best practice, specify a resource using its [Amazon Resource Name \(ARN\)](#). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*" 
```

Important

Lightsail does not support resource-level permissions for some API actions. For more information, see [Support for resource-level permissions and authorization based on tags](#).

The Lightsail instance resource has the following ARN:

```
arn:${Partition}:lightsail:${Region}:${Account}:Instance/${InstanceId}
```

For more information about the format of ARNs, see [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#).

For example, to specify the ea123456-e6b9-4f1d-b518-3ad1234567e6 instance in your statement, use the following ARN:

```
"Resource": "arn:aws:lightsail:us-east-1:123456789012:Instance/ea123456-e6b9-4f1d-b518-3ad1234567e6"
```

To specify all instances that belong to a specific account, use the wildcard (*):

```
"Resource": "arn:aws:lightsail:us-east-1:123456789012:Instance/*"
```

Some Lightsail actions, such as those for creating resources, cannot be performed on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*"
```

Many Lightsail API actions involve multiple resources. For example, `AttachDisk` attaches a Lightsail block storage disk to an instance, so an IAM user must have permissions to use the disk and the instance. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [  
    "resource1",  
    "resource2"
```

To see a list of Lightsail resource types and their ARNs, see [Resources Defined by Amazon Lightsail](#) in the *IAM User Guide*. To learn with which actions you can specify the ARN of each resource, see [Actions Defined by Amazon Lightsail](#).

Condition Keys

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or *Condition block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use [condition operators](#), such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see [IAM policy elements: variables and tags](#) in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

Lightsail does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see [AWS Global Condition Context Keys](#) in the *IAM User Guide*.

To see a list of Lightsail condition keys, see [Condition Keys for Amazon Lightsail](#) in the *IAM User Guide*. To learn with which actions and resources you can use a condition key, see [Actions Defined by Amazon Lightsail](#).

Examples

To view examples of Lightsail identity-based policies, see [Amazon Lightsail Identity-Based Policy Examples](#).

Lightsail Resource-Based Policies

Lightsail does not support resource-based policies.

Access Control Lists (ACLs)

Lightsail does not support Access Control Lists (ACLs).

Authorization Based on Lightsail Tags

You can attach tags to Lightsail resources or pass tags in a request to Lightsail. To control access based on tags, you provide tag information in the [condition element](#) of a policy using the `lightsail:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

Important

Lightsail does not support authorization based on tags for some API actions. For more information, see [Support for resource-level permissions and authorization based on tags](#).

For more information about tagging Lightsail resources, see [Tags](#).

To view an example identity-based policy for limiting access to a resource based on the tags on that resource, see [Allowing Creation and Deletion of Lightsail Resources Based on Tags](#).

Lightsail IAM Roles

An [IAM role](#) is an entity within your AWS account that has specific permissions.

Using Temporary Credentials with Lightsail

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as [AssumeRole](#) or [GetFederationToken](#).

Lightsail supports using temporary credentials.

Service-Linked Roles

[Service-linked roles](#) allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

Lightsail supports service-linked roles. For details about creating or managing Lightsail service-linked roles, see [Service-linked roles](#).

Service Roles

Lightsail does not support service roles.

Topics

- [Amazon Lightsail identity-based policy examples](#)
- [Amazon Lightsail resource-level permissions policy examples](#)
- [Use service-linked roles for Amazon Lightsail](#)
- [IAM policy to manage buckets in Amazon Lightsail](#)

Amazon Lightsail identity-based policy examples

By default, IAM users and roles don't have permission to create or modify Lightsail resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see [Creating Policies on the JSON Tab](#) in the *IAM User Guide*.

Policy Best Practices

Identity-based policies determine whether someone can create, access, or delete Amazon Lightsail resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see [AWS managed policies](#) or [AWS managed policies for job functions](#) in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see [Policies and permissions in IAM](#) in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see [IAM JSON policy elements: Condition](#) in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see [IAM Access Analyzer policy validation](#) in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see [Configuring MFA-protected API access](#) in the *IAM User Guide*.

For more information about best practices in IAM, see [Security best practices in IAM](#) in the *IAM User Guide*.

Using the Lightsail Console

To access the Amazon Lightsail console, you must have full-access permission to all Lightsail actions and resources. These permissions must allow you to list and view details about the Lightsail resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions (i.e., that is not full-access), the console won't function as intended for entities (IAM users or roles) with that policy.

To ensure that those entities can use the Lightsail console, attach the following policy to the entities. For more information, see [Adding Permissions to a User](#) in the *IAM User Guide*:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lightsail:*"
      ],
      "Resource": "*"
    }
  ]
}
```

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're trying to perform.

Allow Users to View Their Own Permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
```



```

        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

Allowing Creation and Deletion of Lightsail Resources Based on Tags

You can use conditions in your identity-based policy to control access to Lightsail resources based on tags. This example shows how you might create a policy that restricts users from creating new Lightsail resources unless a key tag of `allow` and a value of `true` is defined with the create request. This policy also restricts users from deleting resources unless they have the `allow/true` key-value tag.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lightsail:Create*",
                "lightsail:TagResource",
                "lightsail:UntagResource"
            ]
        }
    ]
}

```

```

    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/allow": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "lightsail:Delete*",
        "lightsail:TagResource",
        "lightsail:UntagResource"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/allow": "true"
        }
    }
}
]
}

```

The following example restricts users from changing the tag for resources that have a key-value tag that is not allow/false.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "lightsail:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceTag/allow": "false"
        }
      }
    }
  ]
}

```

```
]
}
```

You can attach these policies to the IAM users in your account. For more information, see [IAM JSON Policy Elements: Condition](#) in the *IAM User Guide*.

Amazon Lightsail resource-level permissions policy examples

The term *resource-level permissions* refers to the ability to specify which resources users are allowed to perform actions on. Amazon Lightsail supports resource-level permissions. This means that for certain Lightsail actions, you can control when users are allowed to use those actions based on conditions that have to be fulfilled, or specific resources that users are allowed to use or edit. For example, you can grant users permissions to manage an instance or database with a specific Amazon Resource Name (ARN).

Important

Lightsail does not support resource-level permissions for some API actions. For more information, see [Support for resource-level permissions and authorization based on tags](#).

For more information about the resources that are created or modified by the Lightsail actions, and the ARNs and Lightsail condition keys that you can use in an IAM policy statement, see [Actions, Resources, and Condition Keys for Amazon Lightsail](#) in the *IAM User Guide*.

Allow management of a specific instance

The following policy grants access to reboot/start/stop an instance, manage instance ports, and create instance snapshots for a specific instance. It also provides read-only access to other instance-related information and resources in the Lightsail account. In the policy, replace *InstanceARN* with the Amazon Resource Name (ARN) of your instance.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lightsail:GetActiveNames",
        "lightsail:GetAlarms",
```

```
"lightsail:GetAutoSnapshots",
"lightsail:GetBlueprints",
"lightsail:GetBundles",
"lightsail:GetCertificates",
"lightsail:GetCloudFormationStackRecords",
"lightsail:GetContactMethods",
"lightsail:GetDisk",
"lightsail:GetDisks",
"lightsail:GetDiskSnapshot",
"lightsail:GetDiskSnapshots",
"lightsail:GetDistributionBundles",
"lightsail:GetDistributionLatestCacheReset",
"lightsail:GetDistributionMetricData",
"lightsail:GetDistributions",
"lightsail:GetDomain",
"lightsail:GetDomains",
"lightsail:GetExportSnapshotRecords",
"lightsail:GetInstance",
"lightsail:GetInstanceAccessDetails",
"lightsail:GetInstanceMetricData",
"lightsail:GetInstancePortStates",
"lightsail:GetInstances",
"lightsail:GetInstanceSnapshot",
"lightsail:GetInstanceSnapshots",
"lightsail:GetInstanceState",
"lightsail:GetKeyPair",
"lightsail:GetKeyPairs",
"lightsail:GetLoadBalancer",
"lightsail:GetLoadBalancerMetricData",
"lightsail:GetLoadBalancers",
"lightsail:GetLoadBalancerTlsCertificates",
"lightsail:GetOperation",
"lightsail:GetOperations",
"lightsail:GetOperationsForResource",
"lightsail:GetRegions",
"lightsail:GetRelationalDatabase",
"lightsail:GetRelationalDatabaseBlueprints",
"lightsail:GetRelationalDatabaseBundles",
"lightsail:GetRelationalDatabaseEvents",
"lightsail:GetRelationalDatabaseLogEvents",
"lightsail:GetRelationalDatabaseLogStreams",
"lightsail:GetRelationalDatabaseMetricData",
"lightsail:GetRelationalDatabaseParameters",
"lightsail:GetRelationalDatabases",
```

```

        "lightsail:GetRelationalDatabaseSnapshot",
        "lightsail:GetRelationalDatabaseSnapshots",
        "lightsail:GetStaticIp",
        "lightsail:GetStaticIps",
        "lightsail:IsVpcPeered"
    ],
    "Resource": "*"
},
{
    "Sid": "VisualEditor2",
    "Effect": "Allow",
    "Action": [
        "lightsail:CloseInstancePublicPorts",
        "lightsail:CreateInstanceSnapshot",
        "lightsail:OpenInstancePublicPorts",
        "lightsail:PutInstancePublicPorts",
        "lightsail:RebootInstance",
        "lightsail:StartInstance",
        "lightsail:StopInstance"
    ],
    "Resource": "InstanceARN"
}
]
}

```

To get the ARN for your instance, use the `GetInstance` Lightsail API action, and specify the name of the instance using the `instanceName` parameter. Your instance ARN will be listed in the results of that action as shown in the following example. For more information, see [GetInstance](#) in the *Amazon Lightsail API Reference*.

```

C:\>aws lightsail get-instance --instance-name WordPress-1
{
  "instance": {
    "name": "WordPress-1",
    "arn": "arn:aws:lightsail:us-west-2:138-:Instance/1361427a-3982-98c5-5591fcd",
    "supportCode": "021-202/1-0-113",
    "createdAt": 1581469097.179,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "resourceType": "Instance",
    "tags": [],
    "blueprintId": "wordpress",
    "blueprintName": "WordPress",
    "bundleId": "nano_2_0",
    "addons": [

```

Allow management of a specific database

The following policy grants access to reboot/start/stop and update a specific database. It also provides read-only access to other database-related information and resources in the Lightsail account. In the policy, replace *DatabaseARN* with the Amazon Resource Name (ARN) of your database.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lightsail:GetActiveNames",
        "lightsail:GetAlarms",
        "lightsail:GetAutoSnapshots",
        "lightsail:GetBlueprints",
        "lightsail:GetBundles",
        "lightsail:GetCertificates",
        "lightsail:GetCloudFormationStackRecords",
        "lightsail:GetContactMethods",
        "lightsail:GetDisk",
        "lightsail:GetDisks",
        "lightsail:GetDiskSnapshot",
        "lightsail:GetDiskSnapshots",
        "lightsail:GetDistributionBundles",
        "lightsail:GetDistributionLatestCacheReset",
        "lightsail:GetDistributionMetricData",
        "lightsail:GetDistributions",
        "lightsail:GetDomain",
        "lightsail:GetDomains",
        "lightsail:GetExportSnapshotRecords",
        "lightsail:GetInstance",
        "lightsail:GetInstanceAccessDetails",
        "lightsail:GetInstanceMetricData",
        "lightsail:GetInstancePortStates",
        "lightsail:GetInstances",
        "lightsail:GetInstanceSnapshot",
        "lightsail:GetInstanceSnapshots",
        "lightsail:GetInstanceState",
        "lightsail:GetKeyPair",
        "lightsail:GetKeyPairs",
```

```

        "lightsail:GetLoadBalancer",
        "lightsail:GetLoadBalancerMetricData",
        "lightsail:GetLoadBalancers",
        "lightsail:GetLoadBalancerTlsCertificates",
        "lightsail:GetOperation",
        "lightsail:GetOperations",
        "lightsail:GetOperationsForResource",
        "lightsail:GetRegions",
        "lightsail:GetRelationalDatabase",
        "lightsail:GetRelationalDatabaseBlueprints",
        "lightsail:GetRelationalDatabaseBundles",
        "lightsail:GetRelationalDatabaseEvents",
        "lightsail:GetRelationalDatabaseLogEvents",
        "lightsail:GetRelationalDatabaseLogStreams",
        "lightsail:GetRelationalDatabaseMetricData",
        "lightsail:GetRelationalDatabaseParameters",
        "lightsail:GetRelationalDatabases",
        "lightsail:GetRelationalDatabaseSnapshot",
        "lightsail:GetRelationalDatabaseSnapshots",
        "lightsail:GetStaticIp",
        "lightsail:GetStaticIps",
        "lightsail:IsVpcPeered"
    ],
    "Resource": "*"
  },
  {
    "Sid": "VisualEditor2",
    "Effect": "Allow",
    "Action": [
      "lightsail:RebootRelationalDatabase",
      "lightsail:StartRelationalDatabase",
      "lightsail:StopRelationalDatabase",
      "lightsail:UpdateRelationalDatabase"
    ],
    "Resource": "DatabaseARN"
  }
]
}

```

To get the ARN for your database, use the `GetRelationalDatabase` Lightsail API action, and specify the name of the database using the `relationalDatabaseName` parameter. Your database ARN will be listed in the results of that action as shown in the following example. For more information, see [GetRelationalDatabase](#) in the *Amazon Lightsail API Reference*.

```
C:\>aws lightsail get-relational-database --relational-database-name Database-1
{
  "relationalDatabase": {
    "name": "Database-1",
    "arn": "arn:aws:lightsail:us-west-2:1381:lightsail:1:RelationalDatabase/3fdf1bef-892c-4c3d-9ccf-10f67",
    "arn:aws:lightsail:us-west-2:1381:lightsail:1:RelationalDatabase/3fdf1bef-892c-4c3d-9ccf-10f67",
    "createdAt": 1576533508.975,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "resourceType": "RelationalDatabase",
    "tags": [],
    "relationalDatabaseBlueprintId": "mysql_8_0",
    "relationalDatabaseBundleId": "micro_1_0",
    "masterDatabaseName": "dbmaster",
    "hardware": {
```

Use service-linked roles for Amazon Lightsail

Amazon Lightsail uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to Amazon Lightsail. Service-linked roles are predefined by Amazon Lightsail and include all the permissions that Lightsail requires to call other AWS services on your behalf.

A service-linked role makes setting up Amazon Lightsail easier because you don't have to manually add the necessary permissions. Amazon Lightsail defines the permissions of its service-linked roles, and unless defined otherwise, only Amazon Lightsail can assume its roles. The defined permissions include the trust policy and the permissions policy, which cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your Amazon Lightsail resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see [AWS Services That Work with IAM](#) and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-Linked Role Permissions for Amazon Lightsail

Amazon Lightsail uses the service-linked role named **AWSServiceRoleForLightsail** – Role to export Lightsail instance and block storage disk snapshots to Amazon Elastic Compute Cloud (Amazon EC2), and to get the current account-level Block Public Access configuration from Amazon Simple Storage Service (Amazon S3).

The AWSServiceRoleForLightsail service-linked role trusts the following services to assume the role:

- `lightsail.amazonaws.com`

The role permissions policy allows Amazon Lightsail to complete the following actions on the specified resources:

- Action: `ec2:CopySnapshot` on all AWS resources.
- Action: `ec2:DescribeSnapshots` on all AWS resources.
- Action: `ec2:CopyImage` on all AWS resources.
- Action: `ec2:DescribeImages` on all AWS resources.
- Action: `cloudformation:DescribeStacks` on all AWS AWS CloudFormation stacks.
- Action: `s3:GetAccountPublicAccessBlock` on all AWS resources.

Service-Linked Role Permissions

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create or edit the description of a service-linked role.

To allow an IAM entity to create a specific service-linked role

Add the following policy to the IAM entity that needs to create the service-linked role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/lightsail.amazonaws.com/AWSServiceRoleForLightsail*",
      "Condition": {"StringLike": {"iam:AWSServiceName": "lightsail.amazonaws.com"}}
    },
    {
      "Effect": "Allow",
      "Action": "iam:PutRolePolicy",
      "Resource": "arn:aws:iam::*:role/aws-service-role/lightsail.amazonaws.com/AWSServiceRoleForLightsail*"
    }
  ]
}
```

```
}
```

To allow an IAM entity to create any service-linked role

Add the following statement to the permissions policy for the IAM entity that needs to create a service-linked role, or any service role that includes the needed policies. This policy attaches a policy to the role.

```
{
  "Effect": "Allow",
  "Action": "iam:CreateServiceLinkedRole",
  "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

To allow an IAM entity to edit the description of any service roles

Add the following statement to the permissions policy for the IAM entity that needs to edit the description of a service-linked role, or any service role.

```
{
  "Effect": "Allow",
  "Action": "iam:UpdateRoleDescription",
  "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

To allow an IAM entity to delete a specific service-linked role

Add the following statement to the permissions policy for the IAM entity that needs to delete the service-linked role.

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
}
```

```
"Resource": "arn:aws:iam::*:role/aws-service-role/lightsail.amazonaws.com/
AWSServiceRoleForLightsail*"
}
```

To allow an IAM entity to delete any service role

Add the following statement to the permissions policy for the IAM entity that needs to delete a service-linked role, or any service-role.

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

Alternatively, you can use an AWS managed policy to provide full access to the service.

Creating a Service-Linked Role for Amazon Lightsail

You don't need to manually create a service-linked role. When you export your Lightsail instance or block storage disk snapshot to Amazon EC2, or create or update a Lightsail bucket in the AWS Management Console, the AWS CLI, or the AWS API, Amazon Lightsail creates the service-linked role for you.

If you delete this service-linked role and need to create it again, you can use the same process to recreate the role in your account. When you export your Lightsail instance or block storage disk snapshot to Amazon EC2, or create or update a Lightsail bucket, Amazon Lightsail creates the service-linked role for you again.

Important

You must configure IAM permissions to allow Amazon Lightsail to create the service-linked role. To do this, complete the steps that are in the following *Service-Linked Role Permissions* section.

Editing a Service-Linked Role for Amazon Lightsail

Amazon Lightsail does not allow you to edit the `AWSServiceRoleForLightsail` service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see [Editing a Service-Linked Role](#) in the *IAM User Guide*.

Deleting a Service-Linked Role for Amazon Lightsail

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must confirm that there are no Amazon Lightsail instance or disk snapshots in a pending copy state before you can delete the `AWSServiceRoleForLightsail` service-linked role. For more information, see [Export snapshots to Amazon EC2](#).

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the `AWSServiceRoleForLightsail` service-linked role. For more information, see [Deleting a Service-Linked Role](#) in the *IAM User Guide*.

Supported Regions for Amazon Lightsail Service-Linked Roles

Amazon Lightsail supports using service-linked roles in all of the regions where the service is available. For more information about the regions that Lightsail is available in, see [Amazon Lightsail Regions](#).

IAM policy to manage buckets in Amazon Lightsail

The following policy grants a user access to manage a specific bucket in the Amazon Lightsail object storage service. This policy grants access to buckets through the Lightsail console, the AWS Command Line Interface (AWS CLI), AWS API, and AWS SDKs. In the policy, replace `<BucketName>` with the name of the bucket to manage. For more information about IAM policies, see [Creating IAM policies](#) in the *AWS Identity and Access Management User Guide*. For more information about creating IAM users and user groups, see [Creating your first IAM delegated user and user group](#) in the *AWS Identity and Access Management User Guide*.

Important

Users who don't have this policy will experience errors when viewing the **Objects** tab of the bucket management page in the Lightsail console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LightsailAccess",
      "Effect": "Allow",
      "Action": "lightsail:*",
      "Resource": "*"
    },
    {
      "Sid": "S3BucketAccess",
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::<BucketName>/*",
        "arn:aws:s3:::<BucketName>"
      ]
    }
  ]
}
```

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
- [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
- [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).

- 11 Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
- 12 Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
- 13 Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
- 14 Learn how to connect your bucket to other resources. For more information, see the following tutorials.
 - [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
- 15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Manage access to Amazon Lightsail for an IAM user

As an [AWS account root user](#), or an AWS Identity and Access Management (IAM) user with administrator access, you can create one or more IAM users in your AWS account, and those users can be configured with different levels of access to services offered by AWS.

For Amazon Lightsail, you might want to create an IAM user who can access only the Lightsail service. You do this when someone joins your team who requires access to view, create, edit, or delete Lightsail resources but doesn't need access to other services offered by AWS. To configure this, you must first create an IAM policy that grants access to Lightsail, then create an IAM group, and attach the policy to the group. You then create IAM users and make them members of the group, which gives them access to Lightsail.

When someone leaves your team, you can remove the user from the Lightsail access group to revoke their access to Lightsail, if for example, they left your team but still work at your company. Or you can delete the user from IAM, if for example, they left your company and will not require access again.

Contents

- [Create an IAM policy for Lightsail access](#)
- [Create an IAM group for Lightsail access and attach the Lightsail access policy](#)

- [Create an IAM user and add the user to the Lightsail access group](#)

Create an IAM policy for Lightsail access

Follow these steps to create an IAM policy for Lightsail access. For more information, see [Creating IAM Policies](#) in the IAM documentation.

1. Sign in to the [IAM console](#).
2. Choose **Policies** in the left navigation pane.
3. Choose **Create Policy**.
4. In the **Create Policy** page, choose the **JSON** tab.



5. Highlight the contents of the text box, and then copy and paste the following policy configuration text.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "lightsail:*"  
      ],  
      "Resource": "*"   
    }  
  ]  
}
```

The result should look like the following example:



```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "lightsail:*"
8       ],
9       "Resource": "*"
10    }
11  ]
12 }
```

This grants access to all Lightsail actions and resources. Actions that require access to other services offered by AWS, such as enabling VPC peering, exporting Lightsail snapshots to Amazon EC2, or creating Amazon EC2 resources using Lightsail, require additional permissions not included in this policy. For more information, see the following guides:

- [Set up Amazon VPC peering to work with AWS resources outside of Amazon Lightsail](#)
- [Exporting Amazon Lightsail snapshots to Amazon EC2](#)
- [Creating Amazon EC2 instances from exported snapshots in Lightsail](#)

For examples of action-specific and resource-specific permissions that you can grant, see [Amazon Lightsail Resource-Level Permissions Policy Examples](#).

6. Choose **Review Policy**.
7. In the **Review Policy** page, name the policy. Give it a descriptive name; for example, `LightsailFullAccessPolicy`.
8. Add a description, and review the policy settings. If you need to make changes, choose **Previous** to modify the policy.

Review policy

Name*
 Use alphanumeric and '+,=, @, _' characters. Maximum 128 characters.

Description
 Maximum 1000 characters. Use alphanumeric and '+,=, @, _' characters.

Summary

Service	Access level	Resource	Request condition
Allow (1 of 176 services) Show remaining 175			
Lightsail	Full access	All resources	None

9. After you confirm the policy settings are correct, choose **Create Policy**.

The policy is now created and can be added to an existing IAM group, or you can create a new IAM group using the steps in the following section of this guide.

Create an IAM group for Lightsail access and attach the Lightsail access policy

Follow these steps to create an IAM group for Lightsail access, then attach the Lightsail access policy created in the previous section of this guide. For more information, see [Creating IAM Groups](#) and [Attaching a Policy to an IAM Group](#) in the IAM documentation.

1. In the [IAM console](#), choose **Groups** in the left navigation pane.
2. Choose **Create New Group**.
3. In the **Set Group Name** page, name the group. Give it a descriptive name; for example, LightsailFullAccessGroup.
4. In the **Attach Policy** page, search for the Lightsail policy you created earlier in this guide; for example, LightsailFullAccessPolicy.
5. Add a checkmark next to the policy, then choose **Next step**.
6. Review the group settings. If you need to make changes, choose **Previous** to modify the group policies.
7. After you confirm the group settings are correct, choose **Create Group**.

The group is now created, and users added to the group will have access to Lightsail actions and resources. You can add existing IAM users to the group, or you can create new IAM users using the steps in the following section of this guide.

Create an IAM user and add the user to the Lightsail access group

Follow these steps to create an IAM user and add the user to the Lightsail access group. For more information, see [Creating an IAM User in Your AWS Account](#) and [Adding and Removing Users in an IAM Group](#) in the IAM documentation.

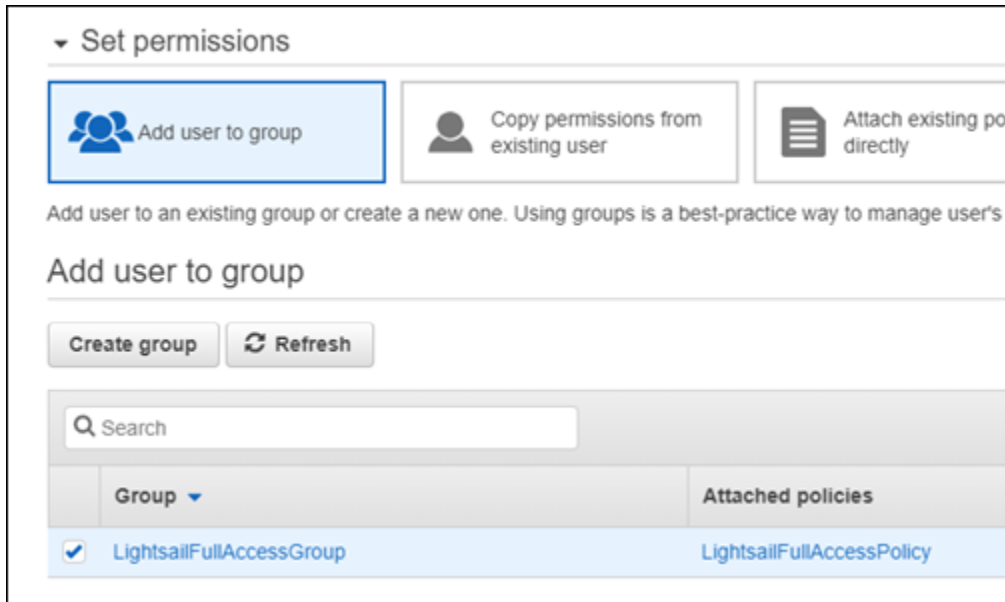
1. In the [IAM console](#), choose **Users** in the left navigation pane.
2. Choose **Add user**.
3. In the **Set user details** section of the page, name the user.
4. Under the **Select AWS access type** section of the page, choose from the following options:
 - a. Choose **Programmatic Access** to enable an access key ID and a secret access key for the AWS API, CLI, SDK, and other development tools, which can be used for Lightsail actions and resources. For more information, see [Configure the AWS CLI to work with Lightsail](#).
 - b. Choose **AWS Management Console access** to enable a password that allows the user to sign in to the AWS Management Console, and thereby the Lightsail console. The following password options appear when this option is selected:
 - i. Choose **Autogenerated password** to have IAM generate the password, or choose **Custom password** to enter your own password.
 - ii. Choose **Require password reset** to have the user create a new password (reset their password) at the next sign in.

Note

If you choose the **Programmatic Access** option only, the user will not be able to sign in to the AWS console, and the Lightsail console.

5. Choose **Next: Permissions**.

6. Under the **Set permissions** section of the page, choose **Add user to group**, and then select the Lightsail access group you created earlier in this guide; for example, `LightsailFullAccessGroup`.



7. Choose **Next: Tags**.
8. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see [Tagging IAM Entities](#).
9. Choose **Next: Review**.
10. Review the user settings. If you need to make changes, choose **Previous** to modify the user's groups or policies.
11. After you confirm the user settings are correct, choose **Create user**.

The user is created, and the user will have access to Lightsail. To revoke the user's Lightsail access, remove the user from the Lightsail access group. For more information, see [Adding and Removing Users in an IAM Group](#) in the IAM documentation.

12. To get the user's credentials, choose the following options:
 - a. Choose **Download .csv** to download a file containing the user name, password, access key ID, secret access key, and the AWS console login link for your account.
 - b. Choose **Show** under **Secret access key** to view the access key that can be used to access Lightsail programmatically (using the AWS API, CLI, SDK, and other development tools).

⚠ Important

This is your only opportunity to view or download the secret access keys, and you must provide this information to your users before they can use the AWS API. Save the user's new access key ID and secret access key in a safe and secure place. You will not have access to the secret keys again after this step.

- c. Choose **Show** under **Password** to view the user's password if it was generated by IAM. You should provide the password to the user so that they can sign in for the first time.
- d. Choose **Send email** to send an email to the user letting them know they now have access to Lightsail.

Success
 You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://138695307491.signin.aws.amazon.com/console>

[Download .csv](#)

	User	Access key ID	Secret access key	Password	Email login instructions
▼	✔ LightsailFull...	AKIAI44QH8DHBIAK13LH43AS2TN7Y6J247S6	***** Show	***** Show	Send email

- ✔ Created user LightsailFullAccessUser
- ✔ Attached policy IAMUserChangePassword to user LightsailFullAccessUser
- ✔ Added user LightsailFullAccessUser to group LightsailFullAccessGroup
- ✔ Created access key for user LightsailFullAccessUser
- ✔ Created login profile for user LightsailFullAccessUser

Update management in Amazon Lightsail

Amazon Web Services (AWS), Amazon Lightsail, and third-party application vendors periodically update and patch the instance images (also known as *blueprints*) that are available on Lightsail. AWS and Lightsail do not update or patch the operating system or applications on instances after you create them. Lightsail also does not update or patch the operating system and software that you configure on your Lightsail container services. Therefore, we recommend that you regularly update, patch, and secure the operating system and applications on your Amazon Lightsail instances and container services. For more information, see the [AWS Shared Responsibility Model](#).

Instance blueprint software support

The following list of Amazon Lightsail platforms and blueprints links to each vendor's support page. There, you can view information such as how-to guides, and keeping your operating system and application up to date. You can use any automatic update service or recommended process for installing updates that are provided by the application vendor.

Windows

- [Windows Server 2022, Windows Server 2019, Windows Server 2016, and Windows Server 2012 R2](#)
- [Microsoft SQL Server](#)

Linux and Unix - Operating system only

- [Amazon Linux 2023](#)
- [Amazon Linux 2](#)
- [Ubuntu](#)
- [Debian](#)
- [FreeBSD](#)
- [openSUSE](#)
- [CentOS](#)

Linux and Unix - Operating system plus application

- [Plesk Hosting Stack on Ubuntu](#)
- [cPanel & WHM for Linux](#)
- [WordPress](#)
- [WordPress Multisite](#)
- [LAMP \(PHP 8\)](#)
- [Node.js](#)
- [Joomla!](#)
- [Magento](#)
- [MEAN](#)

- [Drupal](#)
- [GitLab CE](#)
- [Redmine](#)
- [Nginx](#)
- [Ghost](#)
- [Django](#)
- [PrestaShop](#)

Compliance validation for Amazon Lightsail

AWS provides the following resources to help with compliance:

- [Security and Compliance Quick Start Guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- [AWS Compliance Resources](#) – This collection of workbooks and guides might apply to your industry and location.
- [Evaluating Resources with Rules](#) in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- [AWS Security Hub](#) – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Monitor your Amazon Lightsail resources

Monitor the performance of your instances, databases, distributions, load balancers, container services, and buckets in Amazon Lightsail by checking and collecting their metric data. Establish a baseline over time, so that you can configure alarms to more easily detect anomalies and issues with the performance of your resources.

Amazon Lightsail reports metric data for instances, databases, content delivery network (CDN) distributions, load balancers, container services, and buckets. You can view and monitor this data in the Lightsail console. Monitoring is an important part of maintaining the reliability, availability, and performance of your resources. Monitor and collect metric data from your resources regularly so that you can more readily debug a multi-point failure, if one occurs.

Contents

- [Monitoring your resources effectively](#)
- [Metric concepts and terminology](#)
- [Metrics available in Lightsail](#)

Monitoring your resources effectively

You should establish a baseline for normal resource performance in your environment. Measure performance at various times, and under different load conditions. As you monitor your resources, you should write down and record a history of your resource's performance over time. Compare the current performance of your resources against the historical data that you collected. This helps you identify normal performance patterns and performance anomalies, and devise methods to address them.

For example, you can monitor CPU utilization, network utilization, and status checks for your instances. When performance falls outside your established baseline, you might need to reconfigure or optimize the instance to reduce CPU utilization, or reduce network traffic. If your instance continues to operate above your CPU utilization thresholds, you might want to switch to a larger plan for your instance (e.g., use the \$5 USD/month plan instead of the \$3.50 USD/month plan). You can switch to a larger plan by creating a new snapshot of your instance, and then creating a new instance from the snapshot using the larger plan.

After you've established a baseline, you can configure alarms in the Lightsail console to notify you when your resources cross the specified thresholds. For more information, see [Notifications](#) and [Alarms](#).

Metric concepts and terminology

The following terminology and concepts help you better understand the use of metrics in Lightsail.

Metrics

A metric represents a time-ordered set of data points. Think of a metric as a variable that you monitor, and the data points as representing the values of that variable over time. Metrics are uniquely defined by a name. For example, some instance metrics provided by Lightsail include CPU utilization (`CPUUtilization`), incoming network traffic (`NetworkIn`), and outgoing network traffic (`NetworkOut`). For more information about all of the resource metrics available in Lightsail, see [Metrics available in Lightsail](#).

Metrics retention

Data points with a period of 60 seconds (1 minute resolution) are available for 15 days. Data points with a period of 300 seconds (5 minute resolution) are available for 63 days. Data points with a period of 3600 seconds (1 hour resolution) are available for 455 days (15 months).

Data points that are initially available with a shorter period are aggregated together for long-term storage. For example, data points with a 1 minute granularity remain available for 15 days with 1 minute resolution. After 15 days this data is still available, but is aggregated and is retrievable only with a resolution of 5 minutes. After 63 days, the data is further aggregated and is available with a resolution of 1 hour. If you need availability of metrics longer than these periods, you can use the Lightsail API, AWS Command Line Interface (AWS CLI), and SDKs to retrieve the data points for offline or different storage.

For more information, see [GetInstanceMetricData](#), [GetBucketMetricData](#), [GetLoadBalancerMetricData](#), [GetDistributionMetricData](#), and [GetRelationalDatabaseMetricData](#) in the *Lightsail API reference*.

Statistics

Metric statistics are the means in which data is aggregated over a period of time. Example statistics include Average, Sum, and Maximum. For example, instance CPU utilization metric data can be

averaged using the Average statistic, database connections can be added using the Sum statistic, the maximum load balancer response time can be retrieved using the Maximum statistic, and so on.

For a list of available metric statistics, see [statistics for GetInstanceMetricData](#), [statistics for GetBucketMetricData](#), [statistics for GetLoadBalancerMetricData](#), [statistics for GetDistributionMetricData](#), and [statistics for GetRelationalDatabaseMetricData](#) in the *Lightsail API reference*.

Units

Each statistic has a unit of measure. Example units include Bytes, Seconds, Count, and Percent. For the complete list of the units, see [units for GetInstanceMetricData](#), [units for GetLoadBalancerMetricData](#), [units for GetDistributionMetricData](#), and [units for GetRelationalDatabaseMetricData](#) in the *Lightsail API reference*.

Periods

A period is the length of time associated with a specific data point—the granularity of the returned data points. Each data point represents an aggregation of the metric data collected for a specified period of time. Periods are defined in seconds, and the valid values for period are any multiple of 60 seconds (1-minute) and 300 seconds (5-minutes).

When you retrieve data points using the Lightsail API, you can specify a period, start time, and end time. These parameters determine the overall length of time associated with the data point. Lightsail reports metric data in either 1-minute or 5-minute increments; therefore, you must specify periods in multiples of 60 seconds and 300 seconds. The values that you specify for the start time and end time determine how many periods Lightsail returns. If you prefer statistics aggregated in ten-minute blocks, specify a period of 600. For statistics aggregated over the entire hour, specify a period of 3600, and so on.

Periods are also important for Lightsail alarms. Lightsail evaluates data points for alarms every 5 minutes, and each data point for alarms represents a 5-minute period of aggregated data. When you create an alarm to monitor a specific metric, you are asking Lightsail to compare that metric to the threshold value that you specify. You have extensive control over how Lightsail makes that comparison. You can specify the period over which the comparison is made, and also specify how many evaluation periods are used to reach a conclusion. For more information, see [Alarms](#).

Alarms

An alarm watches a single metric over a specified period of time, and notifies you when the metric crosses a threshold that you specified. The notification can be a banner displayed in the Lightsail console, an email sent to an email address you specified, and a SMS text message sent to a mobile phone number you specified. For more information, see [Alarms](#).

Metrics available in Lightsail

Instance metrics

The following instance metrics are available. For more information, see [Viewing instance metrics in Amazon Lightsail](#).

- **CPU utilization (CPUUtilization)** — The percentage of allocated compute units that are currently in use on the instance. This metric identifies the processing power to run the applications on the instance. Tools in your operating system can show a lower percentage than Lightsail when the instance is not allocated a full processor core.

When viewing the CPU utilization metric graphs for your instances in the Lightsail console, you will see sustainable, and burstable zones. For more information about what these zones mean, see [CPU utilization sustainable and burstable zones](#).

- **Burst capacity minutes (BurstCapacityTime) and percentage (BurstCapacityPercentage)** — Burst capacity minutes represent the amount of time available for your instance to burst at 100% CPU utilization. Burst capacity percentage is the percentage of CPU performance available to your instance. Your instance continuously consumes and accrues burst capacity. Burst capacity minutes are consumed at the full rate only when your instance operates at 100% CPU utilization. For more information about instance burst capacity, see [Viewing instance burst capacity in Amazon Lightsail](#).
- **Incoming network traffic (NetworkIn)** — The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to the instance. The number reported is the number of bytes received during the period. Because this metric is reported in 5-minute intervals, divide the reported number by 300 to find Bytes/second.
- **Outgoing network traffic (NetworkOut)** — The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic from

the instance. The number reported is the number of bytes sent during the period. Because this metric is reported in 5-minute intervals, divide the reported number by 300 to find Bytes/second.

- **Status check failures (`StatusCheckFailed`)** — Reports whether the instance passed or failed both the instance status check and the system status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.
- **Instance status check failures (`StatusCheckFailed_Instance`)** — Reports whether the instance passed or failed the instance status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.
- **System status check failures (`StatusCheckFailed_System`)** — Reports whether the instance passed or failed the system status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.
- **No token metadata requests (`MetadataNoToken`)** — The number of times that the instance metadata service was successfully accessed without a token. This metric determines if there are any processes accessing instance metadata by using Instance Metadata Service Version 1, which doesn't use a token. If all requests use token-backed sessions, such as Instance Metadata Service Version 2, then the value is 0. For more information, see [Instance metadata and user data in Amazon Lightsail](#).

Database metrics

The following database metrics are available. For more information, see [Viewing database metrics in Amazon Lightsail](#).

- **CPU utilization (`CPUUtilization`)** — The percentage of CPU utilization currently in use on the database.
- **Database connections (`DatabaseConnections`)** — The number of database connections in use.
- **Disk queue depth (`DiskQueueDepth`)** — The number of outstanding IOs (read/write requests) that are waiting to access the disk.
- **Free storage space (`FreeStorageSpace`)** — The amount of available storage space.
- **Network receive throughput (`NetworkReceiveThroughput`)** — The incoming (Receive) network traffic on the database, including both customer database traffic and AWS traffic used for monitoring and replication.

- **Network transmit throughput (NetworkTransmitThroughput)** — The outgoing (Transmit) network traffic on the database, including both customer database traffic and AWS traffic used for monitoring and replication.

Distribution metrics

The following distribution metrics are available. For more information, see [Viewing distribution metrics in Amazon Lightsail](#).

- **Requests (Requests)** — The total number of viewer requests received by your distribution, for all HTTP methods, and for both HTTP and HTTPS requests.
- **Bytes uploaded (BytesUploaded)** — The number of bytes uploaded to your origin by your distribution, using POST and PUT requests.
- **Bytes downloaded (BytesDownloaded)** — The number of bytes downloaded by viewers for GET, HEAD, and OPTIONS requests.
- **Total error rate (TotalErrorRate)** — The percentage of all viewer requests for which the response's HTTP status code was 4xx or 5xx.
- **HTTP 4xx error rate (4xxErrorRate)** — The percentage of all viewer requests for which the response's HTTP status code was 4xx. In these cases, the client or client viewer may have made an error. For example, a status code of 404 (Not Found) means that the client requested an object that could not be found.
- **HTTP 5xx error rate (5xxErrorRate)** — The percentage of all viewer requests for which the response's HTTP status code was 5xx. In these cases, the origin server did not satisfy the request. For example, a status code of 503 (Service Unavailable) means that the origin server is currently unavailable.

Load balancer metrics

The following load balancer metrics are available. For more information, see [Viewing load balancer metrics in Amazon Lightsail](#).

- **Healthy host count (HealthyHostCount)** — The number of target instances that are considered healthy.
- **Unhealthy host count (UnhealthyHostCount)** — The number of target instances that are considered unhealthy.

- **Load balancer HTTP 4XX (HTTPCode_LB_4XX_Count)** — The number of HTTP 4XX client error codes that originated from the load balancer. Client errors are generated when requests are malformed or incomplete. These requests were not received by the target instance. This count does not include response codes generated by the target instances.
- **Load balancer HTTP 5XX (HTTPCode_LB_5XX_Count)** — The number of HTTP 5XX server error codes that originated from the load balancer. This does not include any response codes generated by the target instance. This metric is reported if there are no healthy instances attached to the load balancer, or if the request rate exceeds the capacity of the instances (spillover) or the load balancer.
- **Instance HTTP 2XX (HTTPCode_Instance_2XX_Count)** — The number of HTTP 2XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance HTTP 3XX (HTTPCode_Instance_3XX_Count)** — The number of HTTP 3XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance HTTP 4XX (HTTPCode_Instance_4XX_Count)** — The number of HTTP 4XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance HTTP 5XX (HTTPCode_Instance_5XX_Count)** — The number of HTTP 5XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance response time (InstanceResponseTime)** — The time elapsed, in seconds, after the request leaves the load balancer until a response from the target instance is received.
- **Client TLS negotiation error count (ClientTLSNegotiationErrorCount)** — The number of TLS connections initiated by the client that did not establish a session with the load balancer due to a TLS error generated by the load balancer. Possible causes include a mismatch of ciphers or protocols.
- **Request count (RequestCount)** — The number of requests processed over IPv4. This count includes only the requests with a response generated by a target instance of the load balancer.
- **Rejected connection count (RejectedConnectionCount)** — The number of connections that were rejected because the load balancer had reached its maximum number of connections.

Container service metrics

The following container service metrics are available. For more information, see [View container service metrics](#).

- **CPU utilization (CPUUtilization)** — The average percentage of compute units that are currently in use across all nodes of your container service. This metric identifies the processing power required to run containers on your container service.
- **Memory utilization (MemoryUtilization)** — The average percentage of memory that is currently in use across all nodes of your container service. This metric identifies the memory required to run containers on your container service.

Bucket metrics

The following bucket metrics are available. For more information, see [Viewing bucket metrics in Amazon Lightsail](#).

- **Bucket size (BucketSizeBytes)** — The amount of data stored in a bucket. This value is calculated by summing the size of all objects in the bucket (both current and noncurrent objects), including the size of all parts for all incomplete multipart uploads to the bucket.
- **Number of objects (NumberOfObjects)** — The total number of objects stored in a bucket. This value is calculated by counting all objects in the bucket (both current and noncurrent objects) and the total number of parts for all incomplete multipart uploads to the bucket.

Note

Bucket metric data is not reported when your bucket is empty.

Lightsail resource health metrics

You can view the following Amazon Lightsail resource metrics over different time periods. For more information about resource metrics in Lightsail, see [Resource metrics](#).

Instance metrics

The following instance metrics are available. For more information, see [Viewing instance metrics in Amazon Lightsail](#).

- **CPU utilization (CPUUtilization)** — The percentage of allocated compute units that are currently in use on the instance. This metric identifies the processing power to run the applications on the instance. Tools in your operating system can show a lower percentage than Lightsail when the instance is not allocated a full processor core.

When viewing the CPU utilization metric graphs for your instances in the Lightsail console, you will see sustainable, and burstable zones. For more information about what these zones mean, see [CPU utilization sustainable and burstable zones](#).

- **Burst capacity minutes (BurstCapacityTime) and percentage (BurstCapacityPercentage)** — Burst capacity minutes represent the amount of time available for your instance to burst at 100% CPU utilization. Burst capacity percentage is the percentage of CPU performance available to your instance. Your instance continuously consumes and accrues burst capacity. Burst capacity minutes are consumed at the full rate only when your instance operates at 100% CPU utilization. For more information about instance burst capacity, see [View instance burst capacity](#).
- **Incoming network traffic (NetworkIn)** — The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to the instance. The number reported is the number of bytes received during the period. Because this metric is reported in 5-minute intervals, divide the reported number by 300 to find Bytes/second.
- **Outgoing network traffic (NetworkOut)** — The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic from the instance. The number reported is the number of bytes sent during the period. Because this metric is reported in 5-minute intervals, divide the reported number by 300 to find Bytes/second.
- **Status check failures (StatusCheckFailed)** — Reports whether the instance passed or failed both the instance status check and the system status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.
- **Instance status check failures (StatusCheckFailed_Instance)** — Reports whether the instance passed or failed the instance status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.

- **System status check failures (StatusCheckFailed_System)** — Reports whether the instance passed or failed the system status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.
- **System status check failures (StatusCheckFailed_System)** — Reports whether the instance passed or failed the system status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.
- **No token metadata requests (MetadataNoToken)** — The number of times that the instance metadata service was successfully accessed without a token. This metric determines if there are any processes accessing instance metadata by using Instance Metadata Service Version 1, which doesn't use a token. If all requests use token-backed sessions, such as Instance Metadata Service Version 2, the value is 0. For more information, see [Instance metadata and user data](#).

Database metrics

The following database metrics are available. For more information, see [View database metrics](#).

- **CPU utilization (CPUUtilization)** — The percentage of CPU utilization currently in use on the database.
- **Database connections (DatabaseConnections)** — The number of database connections in use.
- **Disk queue depth (DiskQueueDepth)** — The number of outstanding IOs (read/write requests) that are waiting to access the disk.
- **Free storage space (FreeStorageSpace)** — The amount of available storage space.
- **Network receive throughput (NetworkReceiveThroughput)** — The incoming (Receive) network traffic on the database, including both customer database traffic and AWS traffic used for monitoring and replication.
- **Network transmit throughput (NetworkTransmitThroughput)** — The outgoing (Transmit) network traffic on the database, including both customer database traffic and AWS traffic used for monitoring and replication.

Distribution metrics

The following distribution metrics are available. For more information, see [Viewing distribution metrics in Amazon Lightsail](#).

- **Requests** — The total number of viewer requests received by your distribution, for all HTTP methods, and for both HTTP and HTTPS requests.
- **Bytes uploaded** — The number of bytes uploaded to your origin by your distribution, using POST and PUT requests.
- **Bytes downloaded** — The number of bytes downloaded by viewers for GET, HEAD, and OPTIONS requests.
- **Total error rate** — The percentage of all viewer requests for which the response's HTTP status code was 4xx or 5xx.
- **HTTP 4xx error rate** — The percentage of all viewer requests for which the response's HTTP status code was 4xx. In these cases, the client or client viewer may have made an error. For example, a status code of 404 (Not Found) means that the client requested an object that could not be found.
- **HTTP 5xx error rate** — The percentage of all viewer requests for which the response's HTTP status code was 5xx. In these cases, the origin server did not satisfy the request. For example, a status code of 503 (Service Unavailable) means that the origin server is currently unavailable.

Load balancer metrics

The following load balancer metrics are available. For more information, see [View load balancer metrics](#).

- **Healthy host count (HealthyHostCount)** — The number of target instances that are considered healthy.
- **Unhealthy host count (UnhealthyHostCount)** — The number of target instances that are considered unhealthy.
- **Load balancer HTTP 4XX (HTTPCode_LB_4XX_Count)** — The number of HTTP 4XX client error codes that originated from the load balancer. Client errors are generated when requests are malformed or incomplete. These requests were not received by the target instance. This count does not include response codes generated by the target instances.
- **Load balancer HTTP 5XX (HTTPCode_LB_5XX_Count)** — The number of HTTP 5XX server error codes that originated from the load balancer. This does not include any response codes generated by the target instance. This metric is reported if there are no healthy instances attached to the load balancer, or if the request rate exceeds the capacity of the instances (spillover) or the load balancer.

- **Instance HTTP 2XX (HTTPCode_Instance_2XX_Count)** — The number of HTTP 2XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance HTTP 3XX (HTTPCode_Instance_3XX_Count)** — The number of HTTP 3XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance HTTP 4XX (HTTPCode_Instance_4XX_Count)** — The number of HTTP 4XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance HTTP 5XX (HTTPCode_Instance_5XX_Count)** — The number of HTTP 5XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance response time (InstanceResponseTime)** — The time elapsed, in seconds, after the request leaves the load balancer until a response from the target instance is received.
- **Request count (RequestCount)** — The number of requests processed over IPv4. This count includes only the requests with a response generated by a target instance of the load balancer.
- **Client TLS negotiation error count (ClientTLSNegotiationErrorCount)** — The number of TLS connections initiated by the client that did not establish a session with the load balancer due to a TLS error generated by the load balancer. Possible causes include a mismatch of ciphers or protocols.
- **Rejected connection count (RejectedConnectionCount)** — The number of connections that were rejected because the load balancer had reached its maximum number of connections.

Container service metrics

The following container service metrics are available. For more information, see [View container service metrics](#).

- **CPU utilization** — The average percentage of compute units that are currently in use across all nodes of your container service. This metric identifies the processing power required to run containers on your container service.
- **Memory utilization** — The average percentage of memory that is currently in use across all nodes of your container service. This metric identifies the memory required to run containers on your container service.

Bucket metrics

The following bucket metrics are available. For more information, see [View bucket metrics](#).

- **Bucket size** — The amount of data stored in a bucket. This value is calculated by summing the size of all objects in the bucket (both current and non-current objects), including the size of all parts for all incomplete multipart uploads to the bucket.
- **Number of objects** — The total number of objects stored in a bucket. This value is calculated by counting all objects in the bucket (both current and non-current objects) and the total number of parts for all incomplete multipart uploads to the bucket.

Note

Bucket metric data is not reported when your bucket is empty.

Topics

- [Metric notifications in Lightsail](#)
- [View Lightsail instance burst capacity](#)
- [View Lightsail instance metrics](#)
- [Metric alarms in Lightsail](#)
- [Create Lightsail instance metric alarms](#)
- [Delete or disable Lightsail metric alarms](#)

Metric notifications in Lightsail

You can configure Lightsail to notify you when a metric for one of your instances, databases, load balancers, or content delivery network (CDN) distributions crosses a specified threshold. Notifications can be in the form of a banner displayed in the Lightsail console, an email sent to an address you specify, or an SMS text message sent to a mobile phone number you specify.

To get notifications, you must configure an alarm that monitors a metric for one of your resources. For example, you can configure an alarm that notifies you when your instance's outgoing network traffic is greater than 500 kilobytes during a specified length of time. For more information, see [Metric alarms](#).

When an alarm is triggered, a notification banner is displayed in the Lightsail console. To be notified by email and SMS text message, you must add your email address and mobile phone number as notification contacts in each AWS Region where you want to monitor your resources. For more information, see [Add notification contacts](#).

Note

SMS text messaging is not supported in all AWS Regions in which you can create Lightsail resources, and text messages cannot be sent to some countries and regions of the world. For more information, see [Add notification contacts](#).

If you don't receive notifications when you expect to be notified, then there are a few things you should check to confirm that your notification contacts are configured correctly. To learn more, see [Troubleshoot notifications](#).

To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete or disable metric alarms](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

View Lightsail instance burst capacity

Amazon Lightsail offers instances that provide a baseline amount of CPU performance, but also have the ability to temporarily provide additional CPU performance above the baseline as needed. This is referred to as bursting. The baseline performance and ability to burst are governed by the following instance metrics:

- **CPU utilization** – The percentage of allocated compute units that are in use on your instance. This metric identifies the processing power used to run applications on your instance.
- **CPU burst capacity percentage** – The percentage of CPU performance available to your instance.
- **CPU burst capacity minutes** – The amount of time available for your instance to burst at 100% CPU utilization.

In this guide, we show you how to monitor these metrics to maximize the availability of your instance.

Contents

- [Understand baseline CPU performance and burst capacity accrual](#)
- [Identify when your instance bursts](#)
- [Monitor CPU burst capacity](#)
- [Troubleshoot high CPU utilization](#)
- [View instance burst capacity](#)

Understand baseline CPU performance and burst capacity accrual

Lightsail instances continuously earn (at a millisecond-level resolution) a set rate of CPU burst capacity per hour, which is also consumed when your instance's CPU utilization is greater than 0%. The accounting process for whether burst capacity is accrued or consumed also happens at a millisecond-level resolution, so you don't have to worry about overspending CPU burst capacity; a short burst of CPU uses a small fraction of burst capacity.

If your instance uses fewer CPU resources than is required for baseline performance (such as when it is idle), the unspent CPU burst capacity is accrued in the form of CPU burst capacity percentage and minutes. If your instance needs to burst above the baseline performance level, it spends the accrued CPU burst capacity. The more CPU burst capacity that your instance has accrued, the more time it can burst beyond its baseline when more performance is needed.

Baseline CPU performance

The following list outlines the performance baselines for each Lightsail instance plan:

- The **Linux or Unix \$3.50 USD/month** and **Windows \$8 USD/month** (2 vCPUs, 512 MB memory, 30 GB storage) instance plans include a 5% CPU utilization performance baseline.
- The **Linux or Unix \$5 USD/month** and **Windows \$12 USD/month** (2 vCPUs, 1 GB memory, 40 GB storage) instance plans include a 10% CPU utilization performance baseline.
- The **Linux or Unix \$10 USD/month** and **Windows \$20 USD/month** (2 vCPUs, 2 GB memory, 60 GB storage) instance plans include a 20% CPU utilization performance baseline.
- The **Linux or Unix \$20 USD/month** and **Windows \$40 USD/month** (2 vCPUs, 4 GB memory, 80 GB storage) instance plans include a 20% CPU utilization performance baseline.
- The **Linux or Unix \$40 USD /month** and **Windows \$70 USD/month** (2 vCPUs, 8 GB memory, 160 GB storage) instance plans include a 30% CPU utilization performance baseline.
- The **Linux or Unix \$80 USD/month** and **Windows \$120 USD/month** (4 vCPUs, 16 GB memory, 320 GB storage) instance plans include a 40% CPU utilization performance baseline.

- The **Linux/Unix \$160 USD/month** and **Windows \$240 USD/month** (8 vCPUs, 32 GB memory, 640 GB storage) instance plans include a 40% CPU utilization performance baseline.

These performance baselines are per vCPU. The CPU utilization metric graph in the Lightsail console averages the CPU utilization and baseline for instances with more than one vCPU. For example, a Linux or Unix-based \$40 USD/month instance has two vCPUs and an averaged CPU utilization baseline of 30%. Therefore, if:

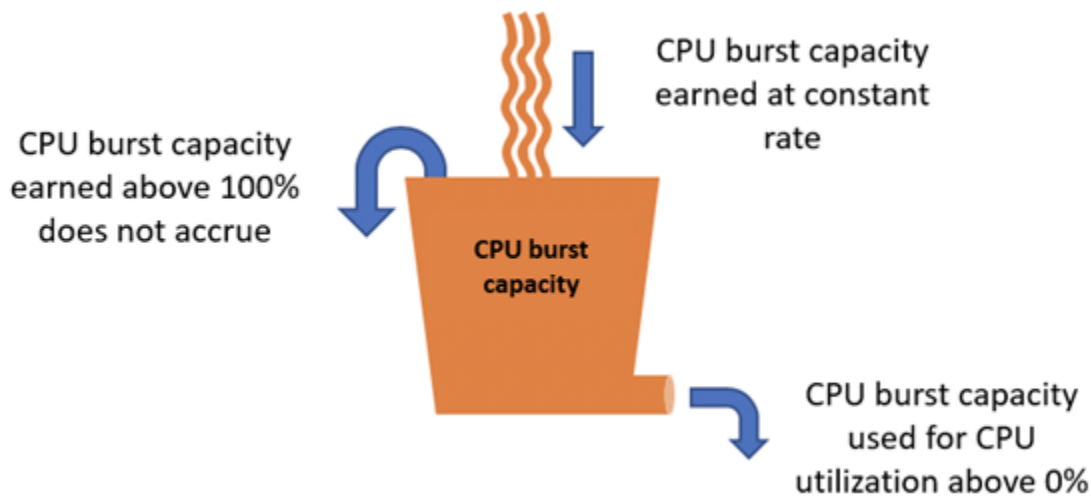
- One vCPU operates at 50% and the other at 0%, a 25% averaged CPU utilization is displayed on the graph. This puts the instance's CPU utilization below its 30% baseline, and in the sustainable zone.
- One vCPU operates at 30%, and the other at 20%, a 25% averaged CPU utilization is displayed on the graph. This puts the instance's CPU utilization below its 30% baseline, and in the sustainable zone.
- One vCPU operates at 35% and the other at 25%, a 30% averaged CPU utilization is displayed on the graph. This puts the instance's CPU utilization at the 30% baseline.
- One vCPU operates at 100% and the other at 90%, a 95% averaged CPU utilization is displayed on the graph. This puts the instance's CPU utilization above its 30% baseline, and in the burstable zone.

Note

For more information about the sustainable and burstable zones, see [Identify when your instance bursts](#) later in this guide.

CPU burst capacity accrual

All Lightsail instance plans accrue 4.17% of CPU burst capacity per hour. The maximum CPU burst capacity that can be accrued is equivalent to the amount of CPU burst capacity percentage that can be earned in a 24-hour period. Your instance stops accruing CPU burst capacity when the CPU burst capacity percentage reaches 100%.



⚠ Important

Accrued CPU burst capacity

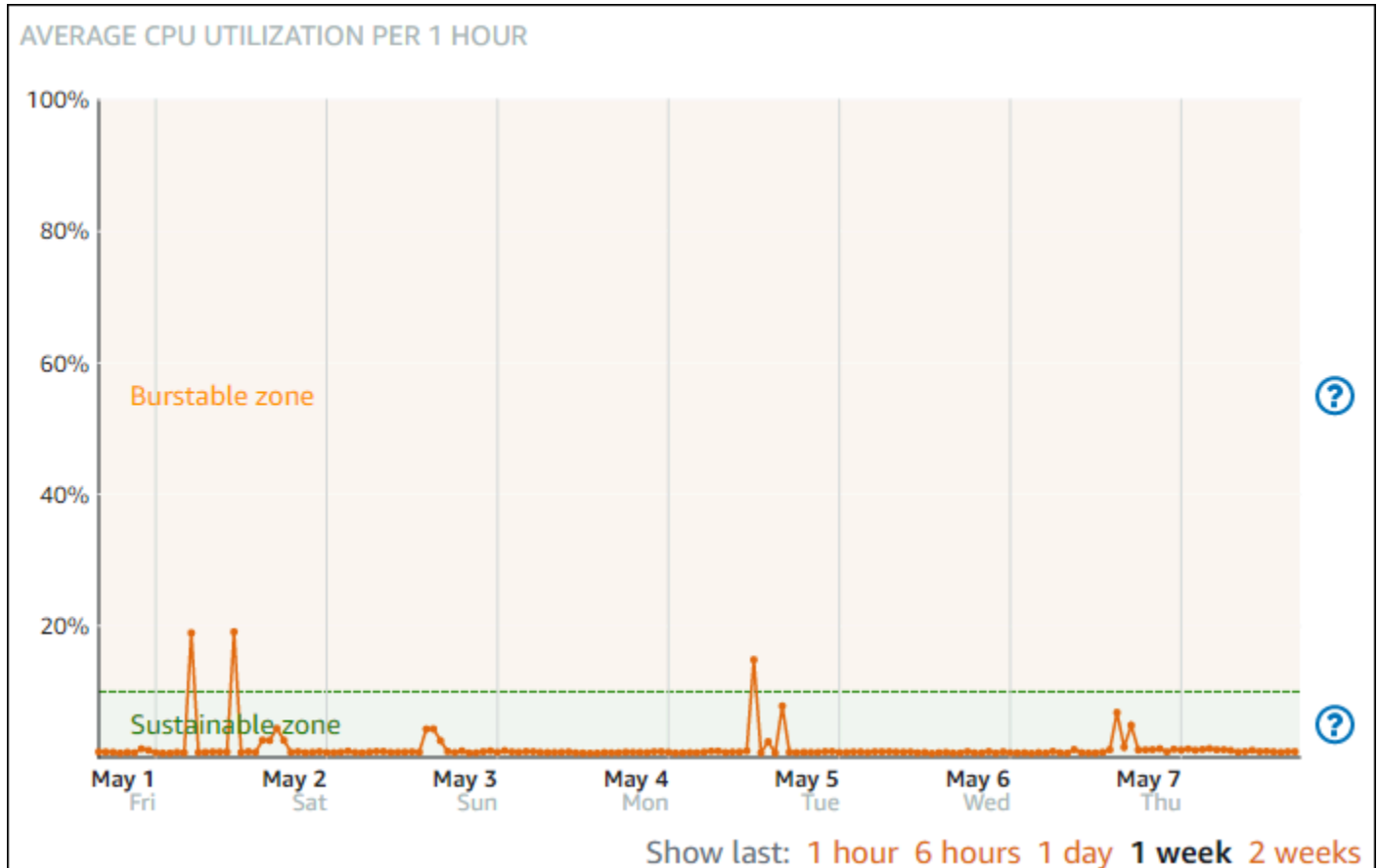
- **Instances created before June 29, 2023**—CPU burst capacity does not persist if your instance is stopped. If you stop your instance, it loses all accrued burst capacity.
- **Instances created on or after June 29, 2023**—CPU burst capacity persists for seven days between instance stops and starts.
- Accrued CPU burst capacity on a running instance does not expire.

Lightsail instances receive additional CPU burst capacity at launch, this is called launch CPU burst capacity. Launch CPU burst capacity allows instances to burst immediately after launch before they have accrued additional burst capacity. Launch CPU burst capacity does not count towards the burst capacity limit. If your instance has not spent its launch CPU burst capacity, and remains idle over a 24-hour period while accruing more burst capacity, its CPU burst capacity (percentage) metric graph will appear as over 100%.

Additionally, some Lightsail instances start in launch mode, which temporarily removes some of the performance limitations that are typically present on burstable instances. Launch mode allows you to run resource-intensive scripts at launch without affecting the overall performance of your instance.

Identify when your instance bursts

On the CPU utilization metric graph for your instances, you will see a sustainable zone, and a burstable zone. In the following CPU utilization metric graph example, the performance baseline is 10% because the instance uses the Linux or Unix-based \$5 USD/month instance plan.

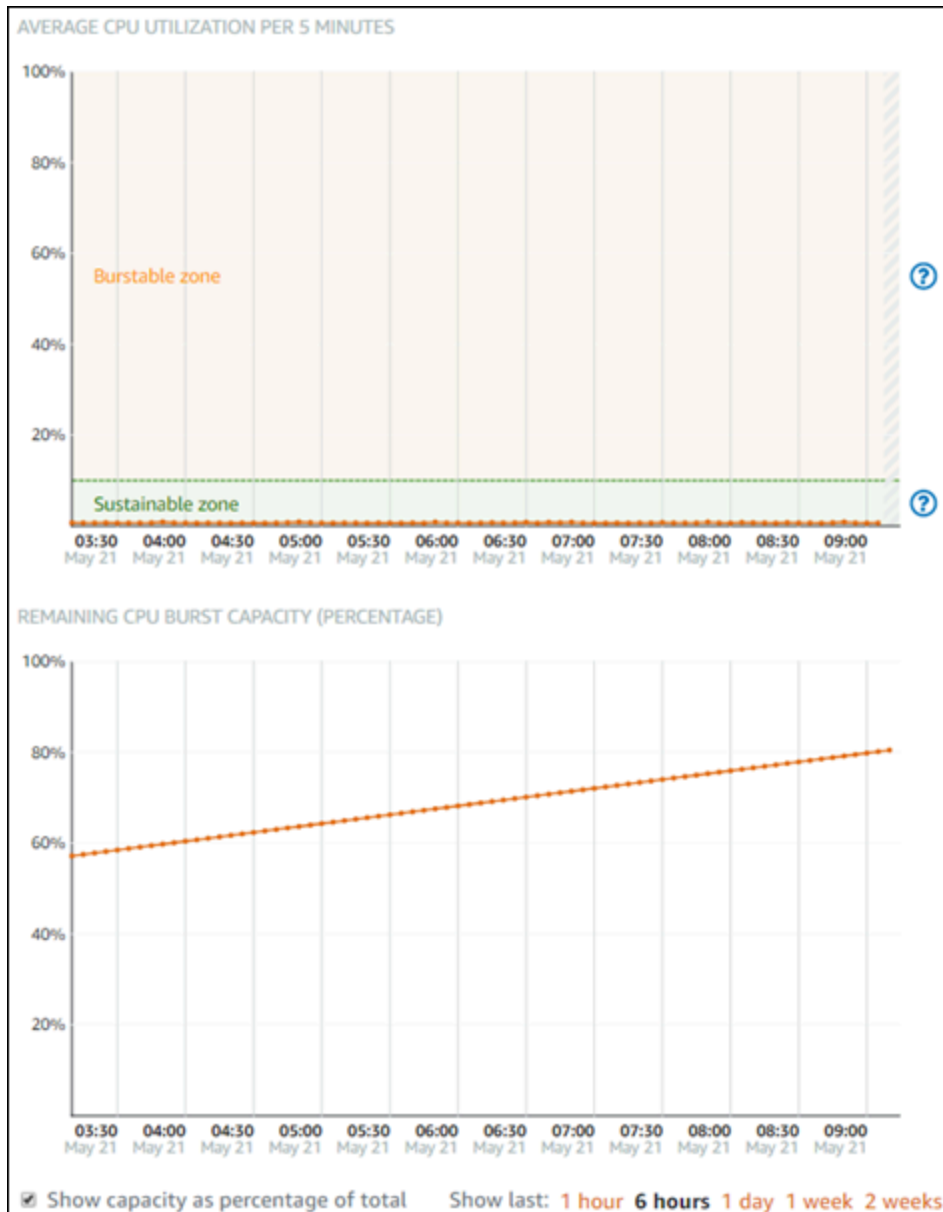


Your Lightsail instance can operate in the sustainable zone indefinitely with no impact to the operation of your system. Your instance may begin operating in the burstable zone when under heavy load, such as when compiling code, installing new software, running a batch job, or serving peak load requests. While operating in the burstable zone, your instance is consuming a higher amount of CPU cycles. Therefore, it can only operate in this zone for a limited period of time.

The period of time your instance can operate in the burstable zone is dependent on how far into the burstable zone it is. An instance operating in the lower end of the burstable zone can burst for a longer period of time than an instance operating in the higher end of the burstable zone. However, an instance that is anywhere in the burstable zone for a sustained period of time will eventually use up all the CPU capacity until it operates in the sustainable zone again. Therefore, it is important to also monitor the remaining CPU burst capacity, which is described in the following section of this guide.

Monitor CPU burst capacity

The CPU overview page in the Lightsail console displays your instance's CPU utilization in comparison to its available CPU burst capacity. In the following CPU overview example, the CPU burst capacity percentage has increased because the instance has continuously operated below its baseline in the sustainable zone.



The remaining CPU burst capacity graph view can be switched between CPU burst capacity percentage and minutes. Your instance consumes more CPU burst capacity when operating in the bursting zone. The CPU burst capacity minutes metric is the amount of time available for your instance to burst at 100% CPU utilization, It is consumed at the same rate as your instance's

current CPU utilization percentage when operating in the burstable zone. For example, a Linux or Unix-based \$5 USD/month instance has a CPU utilization baseline of 10%, and accrues 6 minutes of CPU burst capacity minutes per hour. Therefore, if the instance operates at:

- 100% CPU utilization in the burstable zone for a 60-minute period, then it consumes CPU burst capacity minutes at a 100% rate in that period. The instance consumes 60 minutes of CPU burst capacity, and accrues 6 minutes, for a net consumption of 54 minutes.
- 50% CPU utilization in the burstable zone for a 60-minute period, then it consumes CPU burst capacity minutes at a 50% rate in that period. The instance consumes 30 minutes of CPU burst capacity, and accrues 6 minutes, for a net consumption of 24 minutes.
- 10% CPU utilization at the instance's baseline for a 60-minute period, then it consumes CPU burst capacity minutes at a 10% rate in that period. The instance consumes 6 minutes of CPU burst capacity, and accrues 6 minutes. When an instance operates at its baseline, the CPU burst capacity minutes doesn't increase or decrease.
- 5% CPU utilization in the sustainable zone for a 60-minute period, then it consumes CPU burst capacity minutes at a 5% rate in that period. The instance consumed 3 minutes of CPU burst capacity, and accrued 6 minutes, for a net accrual of 3 minutes.

Alternately, if the instance has accrued 60 minutes of CPU burst capacity, then it can operate at 100% CPU utilization for 60 minutes, at 50% for 120 minutes, or at 25% at 150 minutes.

Troubleshoot high CPU utilization

Your instance will use all of its burst capacity if it operates in the bursting zone frequently, or for extended periods of time. This can signify that your instance is under-provisioned. It could also be that a service is running too frequently, or your instance is running unnecessary software.

Investigate what is causing your instance to burst using tools like `top` on Linux/Unix instances, and Task Manager on Windows Server instances. These tools show you the services that are consuming resources on your instance. Determine which services are consuming the most resources, and identify if they can be disabled without impacting the workload of your instance. By disabling services, or uninstalling software, you may be able to lower the bursting of your instance, and avoid having to up-size your instance.

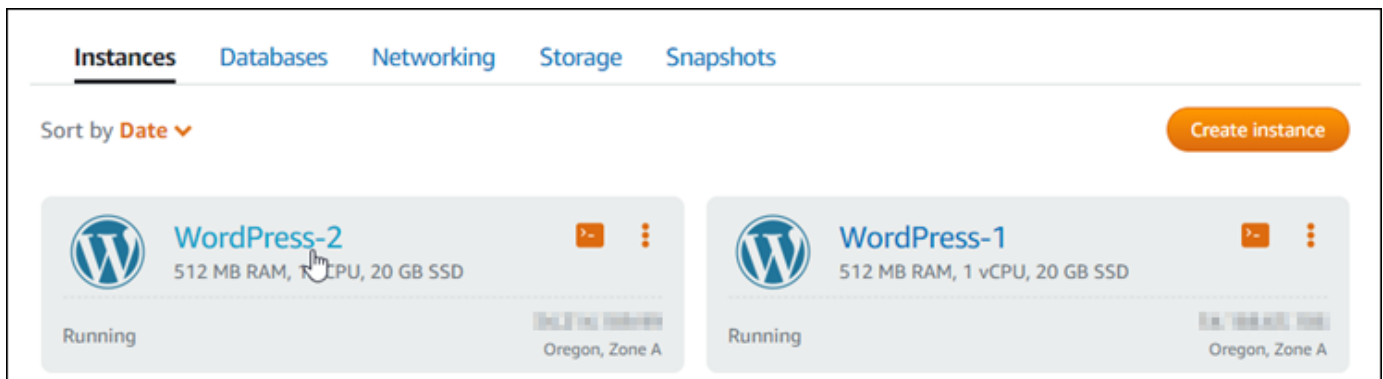
If your instance is truly under-provisioned, and you cannot lower its CPU utilization, then you can mitigate burst capacity consumption by adding more processing power. You do this by creating a snapshot of your instance, and then creating a new instance from the snapshot using a larger

Lightsail instance plan. For example, use the Linux or Unix-based \$20 USD per month plan on your new instance instead of the Linux or Unix-based \$10 USD per month plan used on the previous instance. When your new instance is up and running, make changes to your workload's DNS as necessary to swap the old instance with the new one. Delete your old under-provisioned instance after traffic starts routing to your new instance. For more information, see [Snapshots](#).

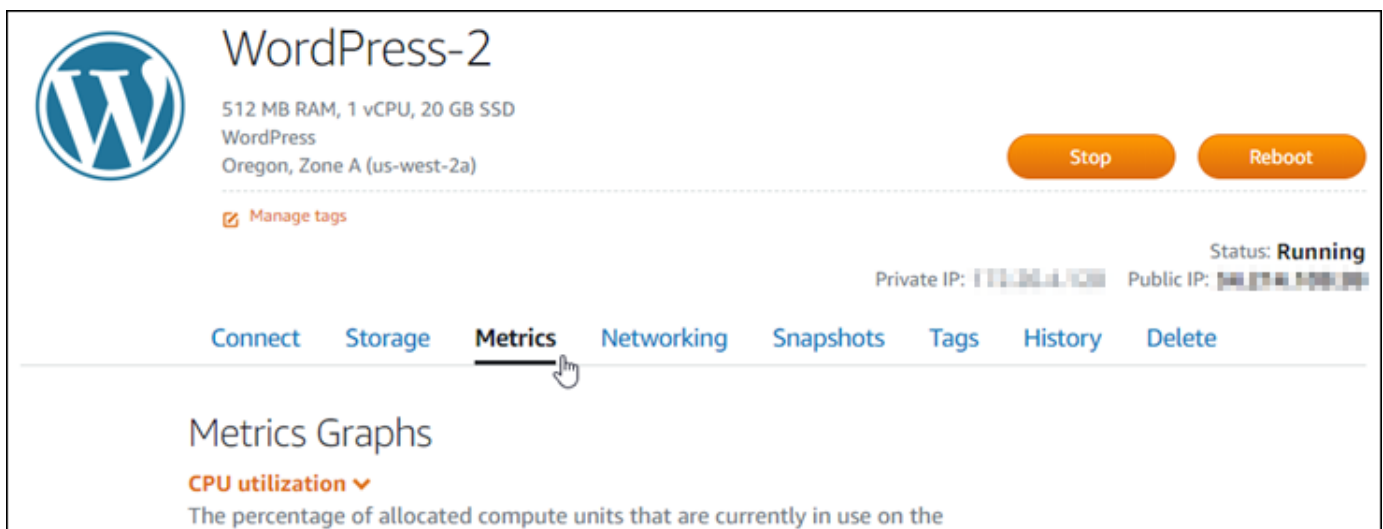
View instance burst capacity

Complete the following steps to access the CPU overview page, and view your instance's CPU utilization and remaining CPU burst capacity.

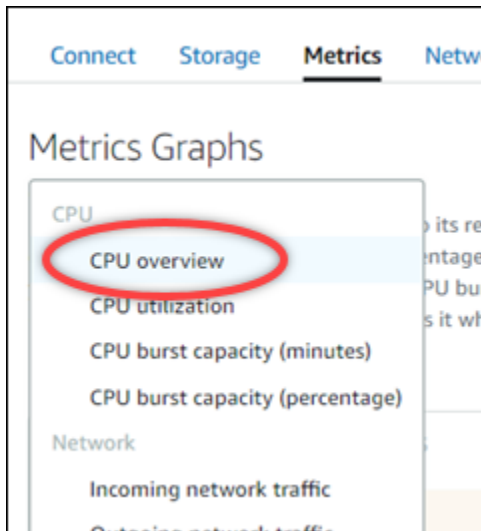
1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.
3. Choose the name of the instance for which you want to view CPU utilization and burst capacity.



4. Choose the **Metrics** tab on the instance management page.



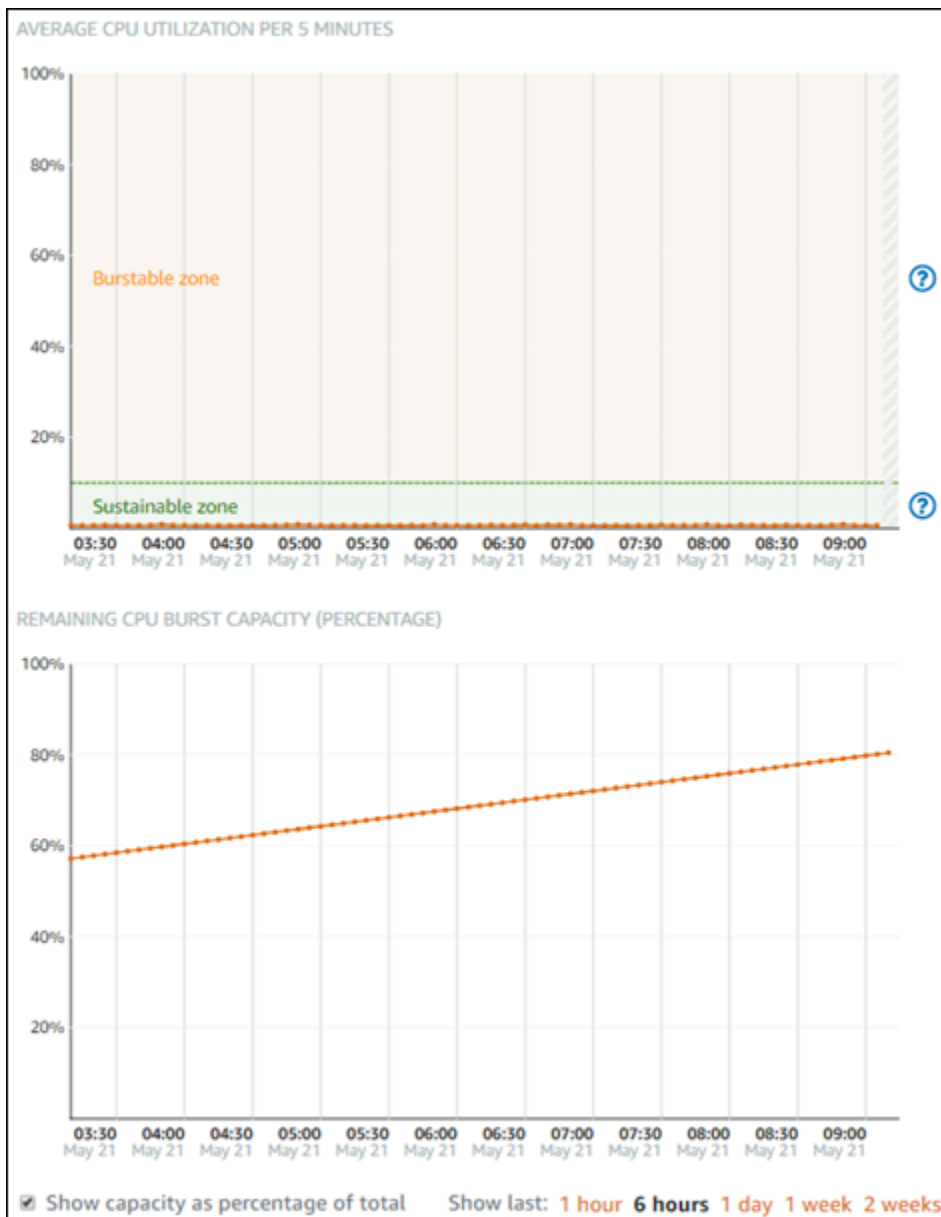
5. Choose **CPU overview** in the drop-down menu under the **Metrics graphs** heading.



The page displays **Average CPU utilization per 5 minutes** and **Remaining CPU burst capacity** graphs.

Note

The **Remaining CPU burst capacity** graph might display a **Launch mode** zone for a short period of time after you create an instance. Some Lightsail instances start in launch mode, which temporarily removes some of the performance limitations that are typically present on burstable instances. Launch mode allows you to run resource-intensive scripts at launch without affecting the overall performance of your instance.



6. You can perform the following actions on the metric graphs:

- For the burst capacity graph, select **Show capacity as percentage of total** to change the view from burst capacity minutes available to burst capacity percentage available.
- Change the view of the graph to show data for 1 hour, 6 hours, 1 day, 1 week, and 2 weeks.
- Pause your cursor on a data point to view detailed information about that data point.
- Add an alarm to be notified when CPU utilization and burst capacity crosses a threshold you specify. Alarms cannot be added in the CPU overview page. You must add them in the

individual CPU utilization, CPU burst capacity percentage, and CPU burst capacity minutes metric graph pages. For more information, see [Alarms](#) and [Create instance metric alarms](#).

View Lightsail instance metrics

After you launch an instance in Amazon Lightsail, you can view its metric graphs on the **Metrics** tab of the instance's management page. Monitoring metrics is an important part of maintaining the reliability, availability, and performance of your resources. Monitor and collect metric data from your resources regularly so that you can more readily debug a multi-point failure, if one occurs. For more information about metrics, see [Metrics in Amazon Lightsail](#).

When monitoring your resources, you should establish a baseline for normal resource performance in your environment. Then you can configure alarms in the Lightsail console to notify you when your resources are performing outside of specified thresholds. For more information, see [Notifications](#) and [Alarms](#).

Contents

- [Instance metrics available in Lightsail](#)
- [CPU utilization sustainable and burstable zones](#)
- [View instance metrics in the Lightsail console](#)
- [Next steps after viewing instance metrics](#)

Available instance metrics

The following instance metrics are available:

- **CPU utilization (CPUUtilization)** — The percentage of allocated compute units that are currently in use on the instance. This metric identifies the processing power to run the applications on the instance. Tools in your operating system can show a lower percentage than Lightsail when the instance is not allocated a full processor core.

When viewing the CPU utilization metric graphs for your instances in the Lightsail console, you will see sustainable, and burstable zones. For more information about what these zones mean, see [CPU utilization sustainable and burstable zones](#).

- **Burst capacity minutes (BurstCapacityTime) and percentage (BurstCapacityPercentage)** — Burst capacity minutes represent the amount of time

available for your instance to burst at 100% CPU utilization. Burst capacity percentage is the percentage of CPU performance available to your instance. Your instance continuously consumes and accrues burst capacity. Burst capacity minutes are consumed at the full rate only when your instance operates at 100% CPU utilization. For more information about instance burst capacity, see [View instance burst capacity](#).

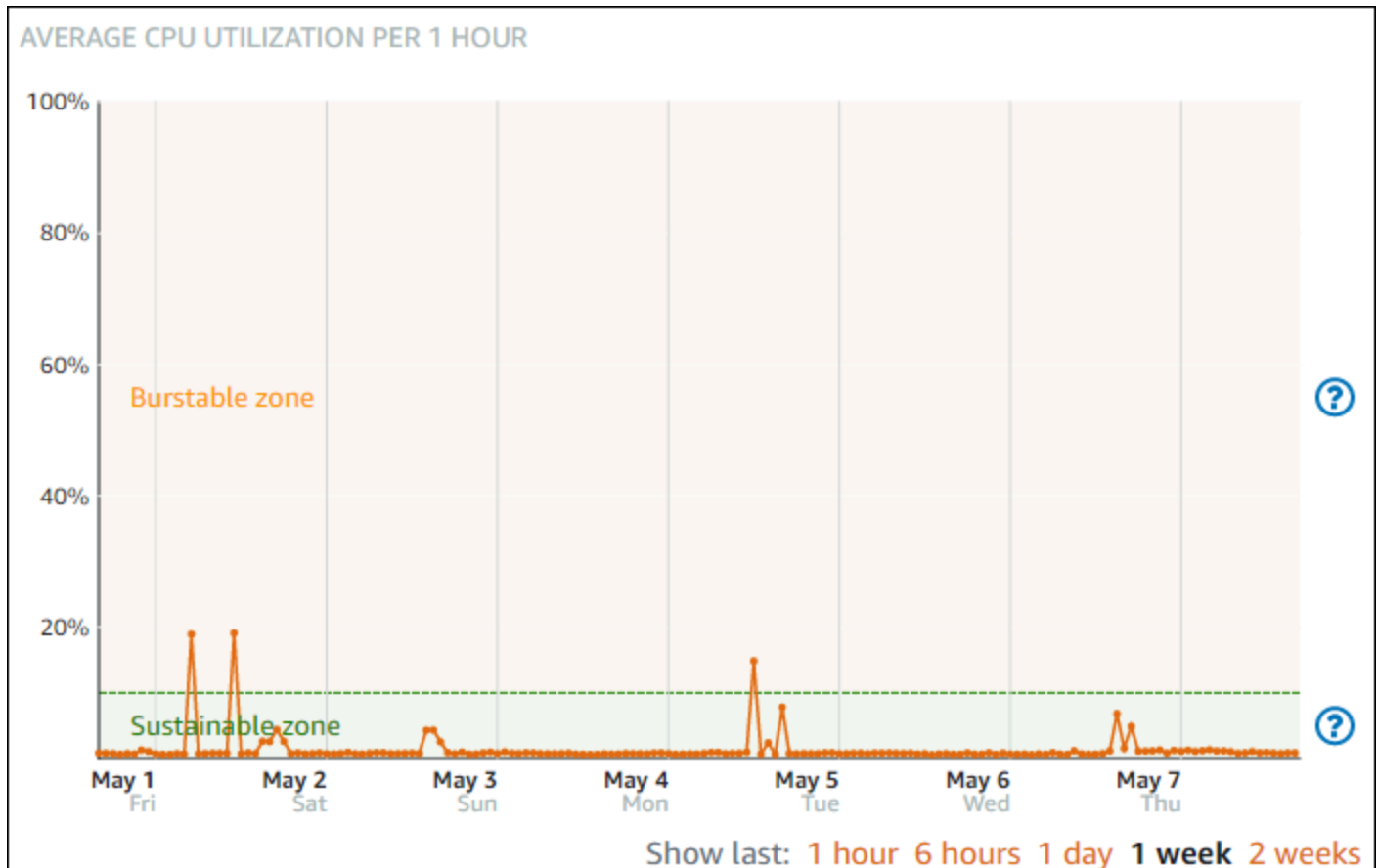
- **Incoming network traffic (NetworkIn)** — The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to the instance. The number reported is the number of bytes received during the period. Because this metric is reported in 5-minute intervals, divide the reported number by 300 to find Bytes/second.
- **Outgoing network traffic (NetworkOut)** — The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic from the instance. The number reported is the number of bytes sent during the period. Because this metric is reported in 5-minute intervals, divide the reported number by 300 to find Bytes/second.
- **Status check failures (StatusCheckFailed)** — Reports whether the instance passed or failed both the instance status check and the system status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.
- **Instance status check failures (StatusCheckFailed_Instance)** — Reports whether the instance passed or failed the instance status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.
- **System status check failures (StatusCheckFailed_System)** — Reports whether the instance passed or failed the system status check. This metric can be either 0 (passed) or 1 (failed). This metric is available at a 1-minute frequency.
- **No token metadata requests (MetadataNoToken)** — The number of times that the instance metadata service was successfully accessed without a token. This metric determines if there are any processes accessing instance metadata by using Instance Metadata Service Version 1, which doesn't use a token. If all requests use token-backed sessions, such as Instance Metadata Service Version 2, then the value is 0. For more information, see [Instance metadata and user data](#).

CPU utilization sustainable and burstable zones

Lightsail uses burstable instances which provide a baseline amount of CPU performance, but also have the ability to temporarily provide additional CPU performance above the baseline as needed. This is referred to as bursting. With burstable instances, you don't have to over-provision your

instance to handle occasional performance spikes—you don't have to pay for capacity you never use.

On the CPU utilization metric graph for your instances, you will see a sustainable zone, and a burstable zone. Your Lightsail instance can operate in the sustainable zone indefinitely with no impact to the operation of your system.



Your instance may begin operating in the burstable zone when under heavy load, such as when compiling code, installing new software, running a batch job, or serving peak load requests. While operating in the burstable zone, your instance is consuming a higher amount of CPU cycles. Therefore, it can only operate in this zone for a limited period of time.

The period of time your instance can operate in the burstable zone is dependent on how far into the burstable zone it is. An instance operating in the lower end of the burstable zone can burst for a longer period of time than an instance operating in the higher end of the burstable zone. However, an instance that is anywhere in the burstable zone for a sustained period of time will eventually use up all the CPU capacity until it operates in the sustainable zone again.

Monitor your instance's CPU utilization metric to see how its performance is distributed between the sustainable and burstable zones. If your system only occasionally moves into the burstable zone, you should be fine continuing to use the instance that you're running. However, if you see your instance spending a considerable amount of time in the burstable zone, you might want to switch to a larger plan for your instance (e.g., use the \$10 USD/month plan instead of the \$3.50 USD/month plan). You can switch to a larger plan by creating a new snapshot of your instance, and then creating a new instance from the snapshot.

View instance metrics in the Lightsail console

Complete the following steps to view instance metrics in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.
3. Choose the name of the instance for which you want to view metrics.
4. Choose the **Metrics** tab on the instance management page.
5. Choose the metric that you want to view in the drop-down menu under the **Metrics graphs** heading.

The graph displays a visual representation of the data points for the chosen metric.

Note

When viewing the CPU utilization metric graphs for your instances in the Lightsail console, you will see sustainable, and burstable zones. For more information about these zones, see [CPU utilization sustainable and burstable zones](#).

6. You can perform the following actions on the metrics graph:
 - Change the view of the graph to show data for 1 hour, 6 hours, 1 day, 1 week, and 2 weeks.
 - Pause your cursor on a data point to view detailed information about that data point.
 - Add an alarm for the chosen metric to be notified when the metric crosses a threshold you specify. For more information, see [Alarms](#) and [Create instance metric alarms](#).

Next steps

There are a few additional tasks that you can perform for your instance metrics:

- Add an alarm for the chosen metric to be notified when the metric crosses a threshold you specify. For more information, see [Metric alarms](#) and [Create instance metric alarms](#).
- When an alarm is triggered, a notification banner is displayed in the Lightsail console. To be notified by email and SMS text message, you must add your email address and mobile phone number as notification contacts in each AWS Region where you want to monitor your resources. For more information, see [Add notification contacts](#).
- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete or disable metric alarms](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

Metric alarms in Lightsail

You can create an alarm in Amazon Lightsail that watches a single metric for your instances, databases, load balancers, and content delivery network (CDN) distributions. The alarm can be configured to notify you based on the value of the metric relative to a threshold that you specify. Notifications can be a banner displayed in the Lightsail console, an email sent to your email address, and an SMS text message sent to your mobile phone number. In this guide, we describe the alarm conditions and settings that you can configure.

Contents

- [Configure an alarm](#)
- [Alarms states](#)
- [Alarm example](#)
- [Configure how alarms treat missing data](#)
- [How alarm state is evaluated when data is missing](#)
- [Missing data in graphed examples](#)
- [More information about alarms](#)

Configuring an alarm

To add an alarm in the Lightsail console, browse to the **Metrics** tab of your instance, database, load balancer, or CDN distribution. You then choose the metric you want to monitor, and choose

Add alarm. You can add two alarms per metric. For more information about metrics, see [Resource metrics](#).

To configure the alarm, you first identify a threshold value, which is the metric value at which point the alarm will change states (e.g., change from an OK state to an ALARM state, or vice versa). For more information, see [Alarms states](#). You then select a comparison operator that will be used to compare the metric to the threshold. The available operators are **greater than or equal to**, **greater than**, **less than**, and **less than or equal to**.

You then specify the number of times the threshold must be crossed, and the period of time the metric will be evaluated, for the alarm to change states. Lightsail evaluates data points for alarms every 5 minutes, and each data point represents a 5 minute period of aggregated data. For example, if you specify the alarm to trigger when the threshold is crossed 2 times, then the evaluation period must be *in the last 10 minutes* or greater (up to 24 hours). If you specify the alarm to trigger when the threshold is crossed 10 times, then the evaluation period must be *in the last 50 minutes* or greater (up to 24 hours).

After you configure the conditions for the alarm, you can configure how you would like to be notified. Notification banners always display in the Lightsail console when the alarm changes from an OK state to an ALARM state. You can also choose to be notified by email and SMS text message, but you must configure notification contacts for those. For more information, see [Metric notifications](#). If you choose to be notified by email and/or SMS text message, you can also choose to be notified when the alarm state changes from an ALARM state to an OK state, which is considered as an *all clear* notification.

Within the **Advanced settings** for the alarm, you can choose how Lightsail treats missing metric data. For more information, see [Configure how alarms treat missing data](#).

Alarms states

An alarm is always in one of the following states:

- **ALARM** — The metric is outside of the defined threshold.

For example, if you choose a **greater than** comparison operator, the alarm will be in an ALARM state when the metric is greater than the specified threshold. If you choose a **less than** comparison operator, the alarm will be in an ALARM state when the metric is less than the specified threshold.

- **OK** — The metric is within the defined threshold.

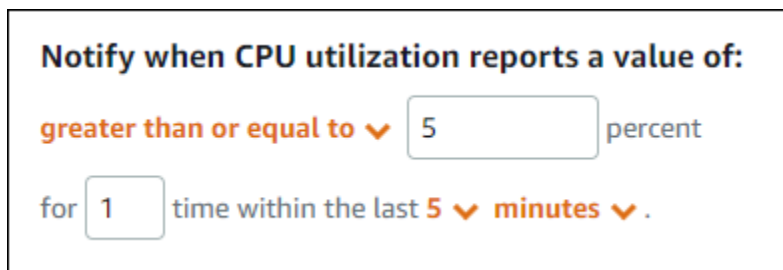
For example, if you choose a **greater than** comparison operator, the alarm will be in an OK state when the metric is less than the specified threshold. If you choose a **less than** comparison operator, the alarm will be in an OK state when the metric is greater than the specified threshold.

- **INSUFFICIENT_DATA** — The alarm has just started, the metric is not available, or there is not enough metric data available for the alarm to determine the alarm state.

Alarms are triggered for state changes only. Alarms are not triggered simply because they are in a particulate state—the state must have changed. When an alarm is triggered, a banner is displayed in the Lightsail console. You can also configure alarms to notify you by email, and SMS text message.

Alarm example

With the previously described alarm conditions in mind, you can configure an alarm that goes into an ALARM state when an instance's CPU utilization is greater than or equal to 5 percent one time in a single 5-minute period. The following example shows the settings for this alarm in the Lightsail console.



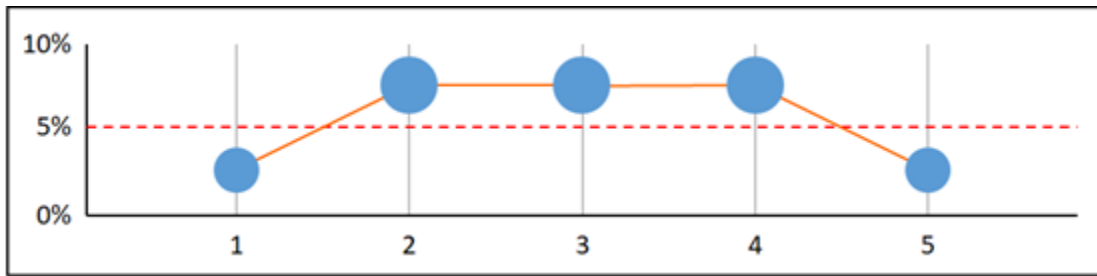
Notify when CPU utilization reports a value of:

greater than or equal to percent

for time within the last minutes.

In this example, if the instance's CPU utilization metric reports a 5 percent or above utilization in just one data point, the alarm changes from an OK state to an ALARM state. Each subsequent data point reported that is 5 percent or above utilization maintains the alarm at an ALARM state. When the instance's CPU utilization metric reports a 4.9 percent or below utilization in just one data point, the alarm changes from an ALARM state to an OK state.

The following graph further illustrates this alarm. The dotted red line represents the 5% CPU utilization threshold, and the blue dots represent metric data points. The alarm is in an OK state for the first data point. The second data point changes the alarm to an ALARM state because the data point is greater than the threshold. The third and fourth data points maintain the ALARM state, because the data points continue to be greater than the threshold. The fifth data point changes the alarm to an OK state because the data point is less than the threshold.



Configure how alarms treat missing data

In some cases, some data points for a metric with an alarm are not reported. For example, this can happen when a connection is lost, or a server goes down.

Lightsail lets you specify how to treat missing data points when configuring an alarm. This helps you configure your alarm to go to the ALARM state when appropriate for the type of data being monitored. You can avoid false positives when missing data doesn't indicate a problem.

Similar to how each alarm is always in one of three states, each specific data point reported falls under one of three categories:

- **Not breaching** — The data point is within the threshold.

For example, if you choose a **greater than** comparison operator, the data point will be **Not breaching** when it is less than the specified threshold. If you choose a **less than** comparison operator, the data point will be **Not breaching** when it is greater than the specified threshold.

- **Breaching** — The data point is outside of the threshold.

For example, if you choose a **greater than** comparison operator, the data point will be **Breaching** when it is greater than the specified threshold. If you choose a **less than** comparison operator, the data point will be **Breaching** when it is less than the specified threshold.

- **Missing** — The behavior for missing data points is specified by the `treat missing data` parameter.

For each alarm, you can specify Lightsail to treat missing data points as any of the following:

- **Not breaching** — Missing data points are treated as "good" and within the threshold.
- **Breaching** — Missing data points are treated as "bad" and breaching the threshold.
- **Ignore** — The current alarm state is maintained.

- **Missing** — The alarm doesn't consider missing data points when evaluating whether to change state. This is the default behavior for alarms.

The best choice depends on the type of metric. For a metric such as an instance's CPU utilization, you might want to treat missing data points as breaching. This is because the missing data points might indicate that something is wrong. But for a metric that generates data points only when an error occurs, such as a load balancer's HTTP 500 server error count, you might want to treat missing data as not breaching.

Choosing the best option for your alarm prevents unnecessary and misleading alarm condition changes. It also more accurately indicates the health of your system.

How alarm state is evaluated when data is missing

No matter what value you set for how to treat missing data, when an alarm evaluates whether to change state, Lightsail attempts to retrieve a greater number of data points than specified by **Evaluation Periods**. The exact number of data points it attempts to retrieve depends on the length of the alarm period. The time frame of the data points that it attempts to retrieve is the evaluation range.

After Lightsail retrieves these data points, the following happens:

- If no data points in the evaluation range are missing, Lightsail evaluates the alarm based on the most recent data points collected.
- If some data points in the evaluation range are missing, but the number of existing data points collected is equal to or more than the alarm's **Evaluation periods**, Lightsail evaluates the alarm state based on the most recent existing data points that were successfully collected. In this case, the value you set for how to treat missing data is not needed, and is then ignored.
- If some data points in the evaluation range are missing, and the number of existing data points that were collected is less than the alarm's number of **Evaluation periods**, Lightsail fills in the missing data points with the result you specified for how to treat missing data, and then evaluates the alarm. However, any real data points in the evaluation range, no matter when they were reported, are included in the evaluation. Lightsail uses missing data points only as few times as possible.

In all of these situations, the number of data points evaluated is equal to the value of **Evaluation periods**. If fewer than the value of **Data points to alarm** are breaching, the alarm state is set to OK. Otherwise, the state is set to ALARM.

Note

A particular case of this behavior is that Lightsail alarms might repeatedly re-evaluate the last set of data points for a period of time after the metric has stopped flowing. This re-evaluation might cause the alarm to change state and re-execute actions, if it had changed state immediately before the metric stream stopping. To mitigate this behavior, use shorter periods.

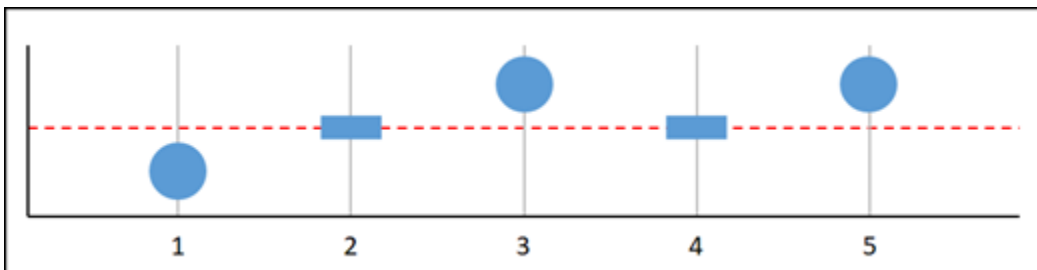
Missing data in graphed examples

The following graphs in this section help illustrate examples of the alarm evaluation behavior. In graphs A, B, C, D, and E, the number data points that must be breaching to alarm, and the evaluation periods, are both 3. The dotted red line represents the threshold, the blue dots represent valid data points, and the dashes represent missing data. Data points above the threshold line are breaching, and data points below the threshold are not breaching. In case some of the most recent three data points are missing, Lightsail will attempt to retrieve additional valid data points.

Note

If data points are missing soon after you create an alarm, and the metric was being reported to Lightsail before you created the alarm, Lightsail retrieves the most recent data points from before the alarm was created when evaluating the alarm.

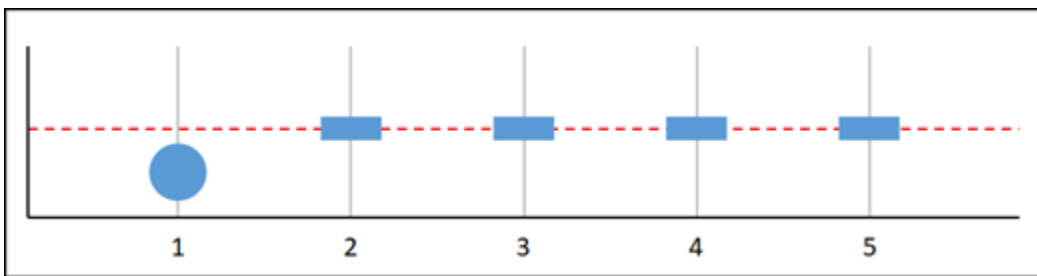
Graph A



In the preceding graphed metric, data point 1 is within threshold, data point 2 is missing, data point 3 is breaching, data point 4 is missing, and data point 5 is breaching. Given that there are three valid data points in the evaluation range, this metric has zero missing data points. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an OK state.
- **Breaching** — The alarm would be in an OK state.
- **Ignore** — The alarm would be in an OK state.
- **Missing** — The alarm would be in an OK state.

Graph B

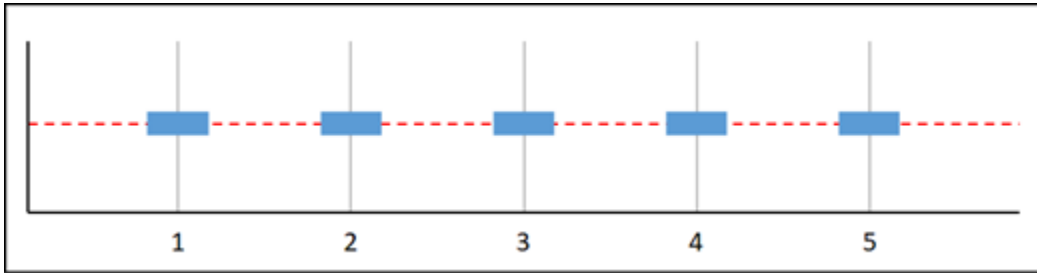


In the preceding graphed metric, data point 1 is within threshold, and data points 2 through 5 are missing. Given that there is only one data point in the evaluation range, this metric has two missing data points. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an OK state.
- **Breaching** — The alarm would be in an OK state.
- **Ignore** — The alarm would be in an OK state.
- **Missing** — The alarm would be in an OK state.

In this scenario, the alarm would stay in an OK state, even if missing data is treated as breaching. This is because the one existing data point is not breaching, and this is evaluated along with two missing data points that are treated as breaching. The next time this alarm is evaluated, if the data is still missing it goes to ALARM. This is because that non-breaching data point is no longer be among the five most recent data points retrieved.

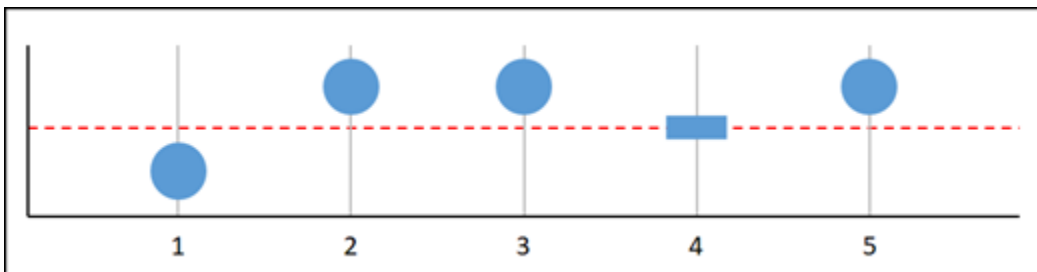
Graph C



All data points are missing in the preceding graphed metric. Given that all data points are missing in the evaluation range, this metric has three missing data points. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an OK state.
- **Breaching** — The alarm would be in an ALARM state.
- **Ignore** — The alarm would maintain the current state.
- **Missing** — The alarm would be in an INSUFFICIENT_DATA state.

Graph D

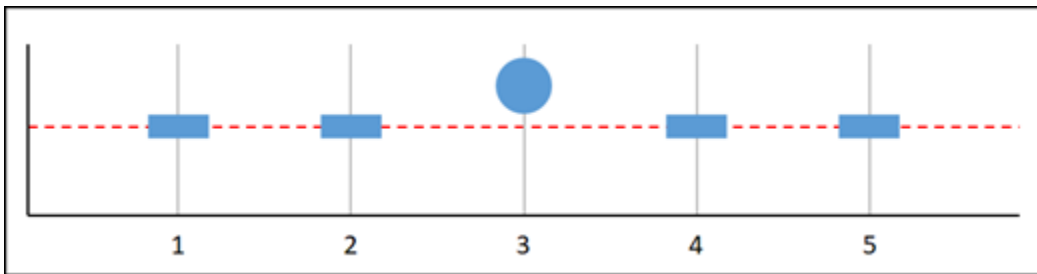


In the preceding graphed metric, data point 1 is within threshold, data point 2 is breaching, data point 3 is breaching, data point 4 is missing, and data point 5 is breaching. Given that there are four valid data points in the evaluation range, this metric has zero missing data points. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an ALARM state.
- **Breaching** — The alarm would be in an ALARM state.
- **Ignore** — The alarm would be in an ALARM state.
- **Missing** — The alarm would be in an ALARM state.

In this scenario, the alarm goes to ALARM state in all cases. This is because there are enough real data points that the setting for how to treat missing data is not needed, and is then ignored.

Graph E

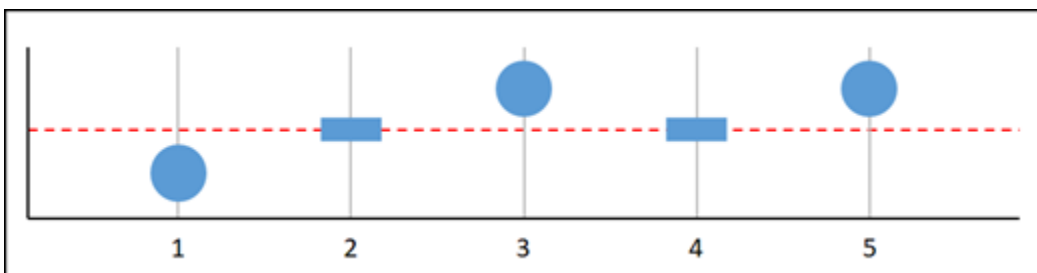


In the preceding graphed metric, data points 1 and 2 are missing, data point 3 is breaching, and data point 4 and 5 are missing. Given that there is only one data point in the evaluation range, this metric has two missing data points. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an OK state.
- **Breaching** — The alarm would be in an ALARM state.
- **Ignore** — The alarm would maintain the current state.
- **Missing** — The alarm would be in an ALARM state.

In graphs F, G, H, I, and J, the **Datapoints to alarm** is 2 while **Evaluation periods** is 3. This is a 2 out of 3, M out of N alarm. 5 is the evaluation range for the alarm.

Graph F

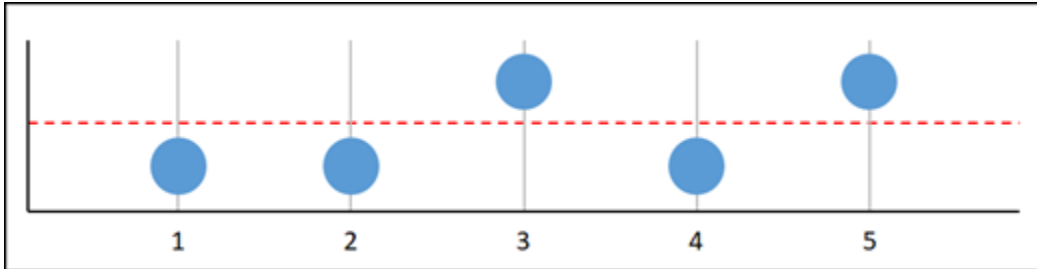


In the preceding graphed metric, data point 1 within threshold, data point 2 is missing, data point 3 is breaching, data point 4 is missing, and data point 5 is breaching. Given that there are three data points in the evaluation range, this metric has zero missing data points. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an ALARM state.

- **Breaching** — The alarm would be in an ALARM state.
- **Ignore** — The alarm would be in an ALARM state.
- **Missing** — The alarm would be in an ALARM state.

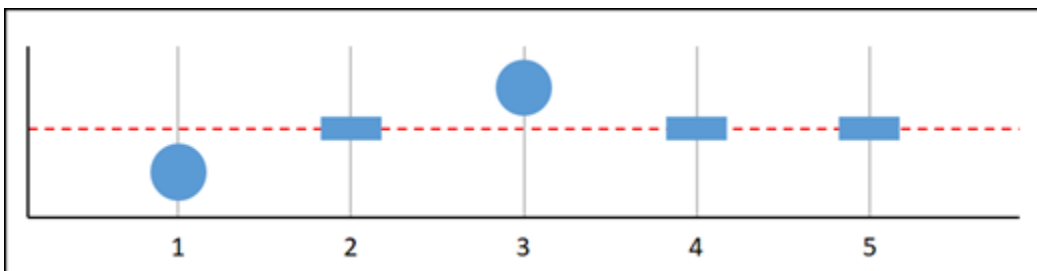
Graph G



In the preceding graphed metric, data points 1 and 2 are within threshold, data point 3 is breaching, data point 4 is within threshold, data point 5 is breaching. Given that there are five data points in the evaluation range, this metric has zero missing data points. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an ALARM state.
- **Breaching** — The alarm would be in an ALARM state.
- **Ignore** — The alarm would be in an ALARM state.
- **Missing** — The alarm would be in an ALARM state.

Graph H

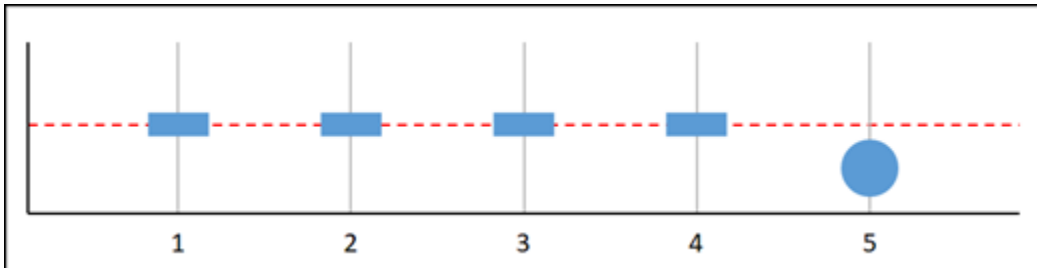


In the preceding graphed metric, data point 1 is within threshold, data point 2 is missing, data point 3 is breaching, and data points 4 and 5 are missing. Given that there are two data points in the evaluation range, this metric has one missing data point. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an OK state.

- **Breaching** — The alarm would be in an ALARM state.
- **Ignore** — The alarm would be in an OK state.
- **Missing** — The alarm would be in an OK state.

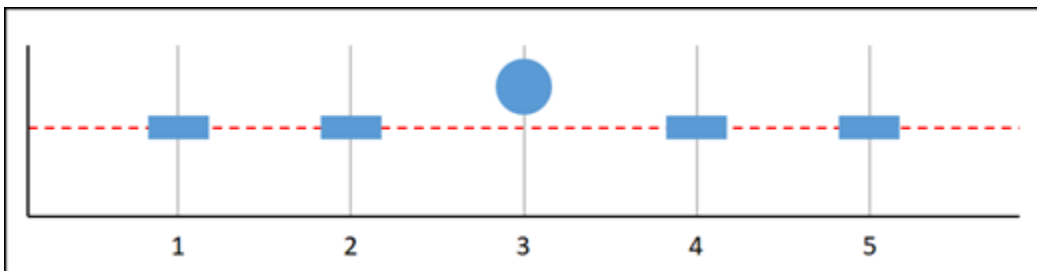
Graph I



In the preceding graphed metric, data points 1 through 4 are missing, and data point 5 is within threshold. Given that there is one data point in the evaluation range, this metric has two missing data points. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an OK state.
- **Breaching** — The alarm would be in an ALARM state.
- **Ignore** — The alarm would be in an OK state.
- **Missing** — The alarm would be in an OK state.

Graph J



In the preceding graphed metric, data points 1 and 2 are missing, data point 3 is breaching, and data point 4 and 5 are missing. Given that there is one data point in the evaluation range, this metric has two missing data points. If you configured an alarm to treat missing data points as:

- **Not breaching** — The alarm would be in an OK state.
- **Breaching** — The alarm would be in an ALARM state.
- **Ignore** — The alarm would maintain the current state.

- **Missing** — The alarm would be in an ALARM state.

More information about alarms

Here are some articles to help you manage alarms in Lightsail:

- [Create instance metric alarms](#)
- [Create database metric alarms](#)
- [Create load balancer metric alarms](#)
- [Create distribution metric alarms](#)
- [Delete or disable metric alarms](#)

Create Lightsail instance metric alarms

You can create an Amazon Lightsail alarm that watches a single instance metric. An alarm can be configured to notify you based on the value of the metric relative to a threshold that you specify. Notifications can be a banner displayed in the Lightsail console, an email sent to your email address, and an SMS text message sent to your mobile phone number. For more information about alarms, see [Alarms](#).

Contents

- [Instance alarm limits](#)
- [Best practices for configuring instance alarms](#)
- [Default alarm settings](#)
- [Create instance metric alarms using the Lightsail console](#)
- [Test instance metric alarms using the Lightsail console](#)
- [Next steps after creating instance alarms](#)

Instance alarm limits

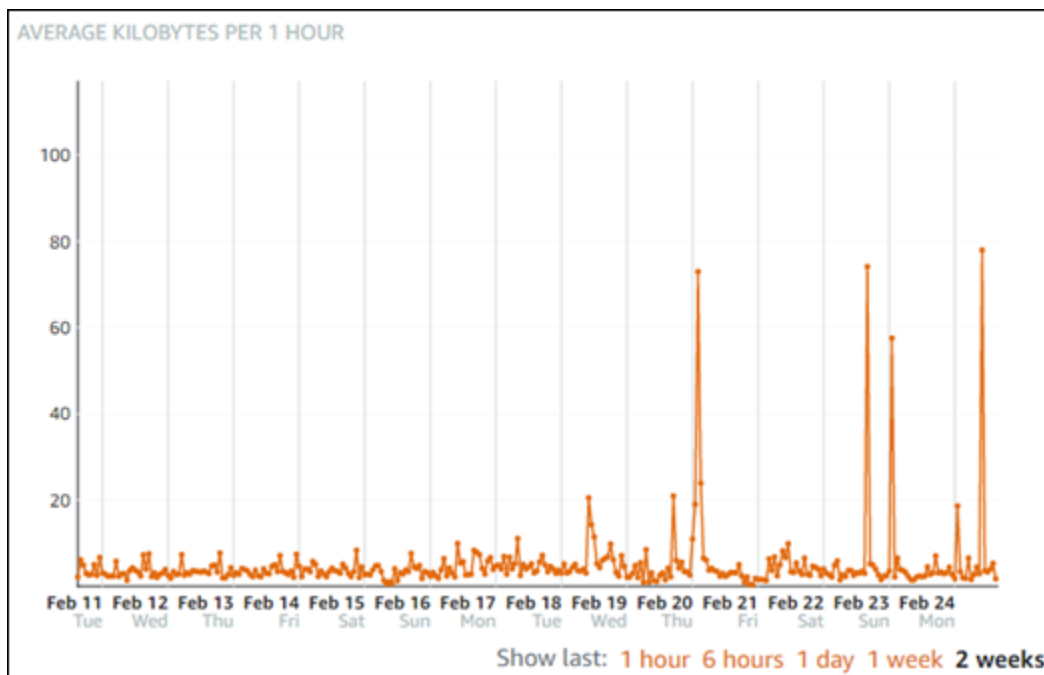
The following limits apply to alarms:

- You can configure two alarms per metric.
- Alarms are evaluated in 5 minute intervals, and each data point for alarms represents a 5 minute period of aggregated metric data.

- You can only configure an alarm to notify you when the alarm state changes to OK if you configure the alarm to notify you by email and/or SMS text message.
- You can only test the OK alarm notification if you configure the alarm to notify you by email and/or SMS text message.
- You can only configure an alarm to notify you when the alarm state changes to INSUFFICIENT_DATA if you configure the alarm to notify you by email and/or SMS text message, and if you choose the **Do not evaluate the missing data** option for missing data points.
- You can only test notifications if the alarm is in an OK state.

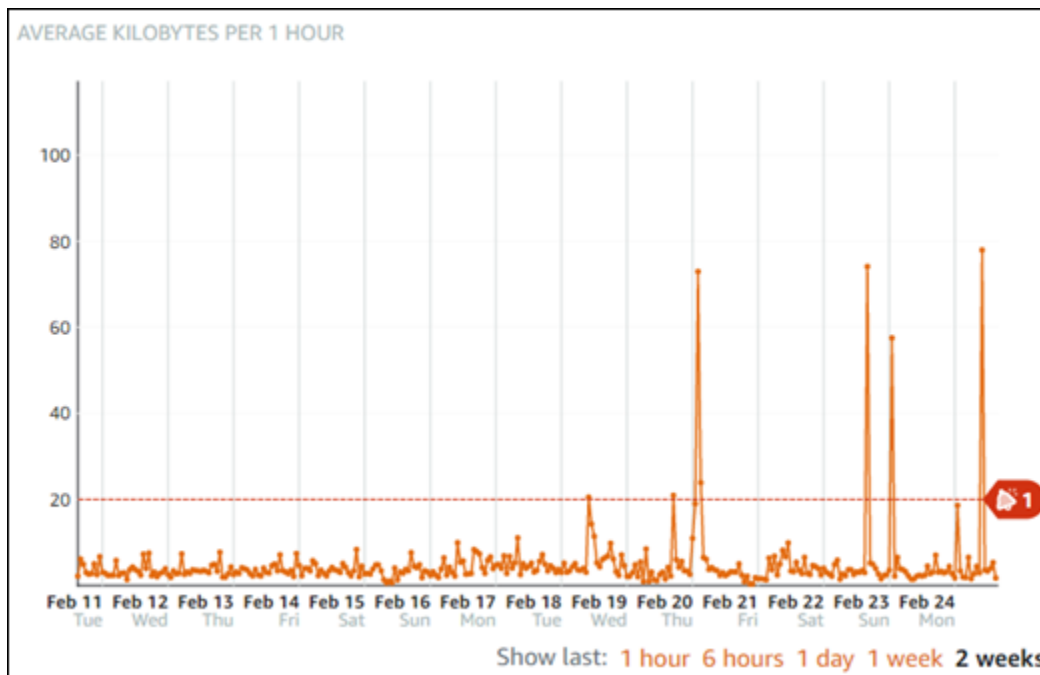
Best practices for configuring instance alarms

Before you configure a metric alarm for your instance, you should view the historical data of the metric. Identify the metric's low-levels, mid-levels, and high-levels over a period of the last two weeks. In the following outgoing network traffic (NetworkOut) metric graph example, the low-levels are 0-10 KB per hour, the mid-levels are between 10-20 KB per hour, and the high-levels are between 20-80 KB per hour.



If you configure the alarm threshold to be **greater than or equal to** somewhere in the low-level range (e.g., 5 KB per hour), then you will get more frequent, and potentially unnecessary alarm notifications. If you configure the alarm threshold to be **greater than or equal to** somewhere in the high-level range (e.g., 20 KB per hour), then you will get less frequent alarm notifications, but that

might be more important to investigate. When you configure an alarm, and enable it, an alarm line representing the threshold appears on the graph as shown in the following example. The alarm line labeled as 1 represents the threshold for Alarm 1, and the alarm line labeled as 2 represents the threshold for Alarm 2.



Default alarm settings


Default alarm settings are prepopulated when you add a new alarm in the Lightsail console. That is the recommended alarm configuration for the metric you selected. However, you should confirm that the default alarm configuration is appropriate for your resource. For example, the default alarm threshold for the instance outgoing network traffic (NetworkOut) metric is **less than or equal to 0 Bytes** for 2 times within the last 10 minutes. However, if you're interested in being notified of a high traffic event, then you might want to modify the alarm threshold to be **greater than or equal to 50 KB** for 2 times within the last 10 minutes, or add a second alarm with these settings so that you're notified when there is no traffic, and when there is high traffic. The threshold that you specify should be adjusted to match the metric high-levels and low-levels as described in the [Best practices for configuring instance alarms](#) section of this guide.

Create instance metric alarms using the Lightsail console

Complete the following steps to create an instance metric alarm using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.

3. Choose the name of the instance for which you want to create alarms.
4. Choose the **Metrics** tab on the instance management page.
5. Choose the metric for which you want to create an alarm in the drop-down menu under the **Metrics Graphs** heading. For more information, see [Resource metrics](#).
6. Choose **Add alarm** in the **Alarms** section of the page.
7. Choose a comparison operator value in the drop-down menu. Example values are greater than or equal to, greater than, less than, or less than or equal to.
8. Enter a threshold for the alarm.
9. Enter the data points to alarm.
10. Choose the evaluation periods. The period can be specified in 5-minute increments, from 5 minutes up to 24 hours.
11. Choose one of the following notification methods:
 - **Email** — You are notified by email when the alarm state changes to ALARM.
 - **SMS text message** — You are notified by SMS text message when the alarm state changes to ALARM. SMS messaging is not supported in all AWS Regions in which you can create Lightsail resources, and SMS text messages cannot be sent to all countries/regions. For more information, see [SMS text messaging support](#).

 **Note**

You are required to add an email address or mobile phone number if you select to be notified by email or SMS but you haven't yet configured a notification contact in the resource's AWS Region. For more information, see [Metric notifications](#).

12. (Optional) Choose **Send me a notification when the alarm state change to OK** to be notified when the alarm state changes to OK. This option is available only if you choose to be notified by Email or SMS text message.
13. (Optional) Choose **Advanced settings**, and then choose one of the following options:
 - Choose how the alarm should treat missing data The following options are available:
 - **Assume it's not within the threshold (Breaching threshold)** — Missing data points are treated as "bad" and breaching the threshold.
 - **Assume it's within the threshold (Not breaching threshold)** — Missing data points are treated as "good" and within the threshold.

- **Use the value of the last good data point (Ignore and maintain the current alarm state)** — The current alarm state is maintained.
- **Do not evaluate it (Treat missing data as missing)** — The alarm doesn't consider missing data points when evaluating whether to change state.
- Choose **Send a notification if there is insufficient data** to be notified when the alarm state changes to `INSUFFICIENT_DATA`. This option is available only if you choose to be notified by Email or SMS text message.

14. Choose **Create** to add the alarm.

To edit the alarm later, choose the ellipsis icon (:) next to the alarm you want to edit, and choose **Edit alarm**.

Test instance metric alarms using the Lightsail console

Complete the following steps to test an alarm using the Lightsail console. You might want to test an alarm to confirm that the configured notification options are working, such as to ensure that you receive an email or an SMS text message when the alarm is triggered.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.
3. Choose the name of the instance for which you want to test an alarm.
4. Choose the **Metrics** tab on the instance management page.
5. Choose the metric for which you want to test an alarm in the drop-down menu under the **Metrics Graphs** heading.
6. Scroll down to the **Alarms** section of the page, and choose the ellipsis icon (:) next to the alarm you want to test.
7. Choose one of the following options:
 - **Test alarm notification** — Choose this option to test the notifications for when the alarm state changes to `ALARM`.
 - **Test OK notification** — Choose this option to test the notifications for when the alarm state changes to `OK`.

Note

If either of these options is unavailable, you might not have configured the notification options for the alarm, or the alarm might currently be in an ALARM state. For more information, see [Instance alarm limits](#).

The alarm momentarily changes to an ALARM or OK state depending on the test option you chose, and an email and/or SMS text message is sent depending on what you configured as the notification method for the alarm. A notification banner displays in the Lightsail console only if you chose to test the ALARM notification. A notification banner is not displayed if you chose to test the OK notification. The alarm will return to its actual state often after a few seconds.

Next steps

There are a few additional tasks that you can perform for your instance alarms:

- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete notification contacts](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

Delete or disable Lightsail metric alarms

You can delete an Amazon Lightsail alarm to stop notifications of when the metric being monitored by the alarm crosses a threshold. You can also disable the alarm to stop receiving notifications. For more information, see [Alarms](#).

Contents

- [Delete metric alarms using the Lightsail console](#)
- [Disable and enable metric alarms using the Lightsail console](#)

Delete metric alarms using the Lightsail console

Complete the following steps to delete a metric alarm using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances**, **Databases**, or **Networking** tab.
3. Choose the name of the resource (instance, database, or load balancer) for which you want to delete an alarm.
4. Choose the **Metrics** tab on the resource's management page.
5. Choose the metric for which you want to delete an alarm in the drop-down under the **Metrics Graphs** heading.
6. Scroll down to the **Alarms** section of the page, and choose the ellipsis icon (:) next to the alarm you want to delete.
7. Choose **Delete**.
8. At the prompt, choose **Delete** to confirm that you want to delete the alarm.

Disable and enabling metric alarms using the Lightsail console

Complete the following steps to disable a metric alarm using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances**, **Databases**, or **Networking** tab.
3. Choose the name of the resource (instance, database, or load balancer) for which you want to disable an alarm.
4. Choose the **Metrics** tab on the resource's management page.
5. Choose the metric for which you want to disable an alarm in the drop-down under the **Metrics Graphs** heading.
6. Scroll down to the **Alarms** section of the page, locate the alarm you want to disable, and choose the toggle to disable it. Likewise, choose the toggle to enable it if it's disabled.

View Lightsail bucket metrics

After you create a bucket in the Amazon Lightsail object storage service, you can view its metric graphs on the **Metrics** tab of the bucket's management page. Monitoring metrics is an important part of maintaining the availability, and performance of your bucket. Monitor and collect metric

data from your bucket regularly so that you can upsize or downsize the storage space and network transfer quota of your bucket when you need to. For more information about metrics, see [Resource metrics](#).

When monitoring your resources, you should establish a baseline for normal resource performance in your environment. Then you can configure alarms in the Lightsail console to notify you when your resources are performing outside of specified thresholds. For more information, see [Notifications](#) and [Alarms](#).

Bucket metrics

The following bucket metrics are available:

- **Bucket size** — The amount of data stored in a bucket. This value is calculated by summing the size of all objects in the bucket (both current and noncurrent objects), including the size of all parts for all incomplete multipart uploads to the bucket.
- **Number of objects** — The total number of objects stored in a bucket. This value is calculated by counting all objects in the bucket (both current and noncurrent objects) and the total number of parts for all incomplete multipart uploads to the bucket.

Note

Bucket metric data is not reported when your bucket is empty.

View bucket metrics in the Lightsail console

Complete the following procedure to view bucket metrics in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to view metrics.
4. Choose the **Metrics** tab on the bucket management page.
5. Choose the metric that you want to view in the dropdown menu under the **Metrics graphs** heading.

The graph displays a visual representation of the data points for the chosen metric.

Screenshot TBD

You can perform the following actions on the metrics graph:

- Change the view of the graph to show data for 1 hour, 6 hours, 1 day, 1 week, and 2 weeks.
- Pause your cursor on a data point to view detailed information about that data point.
- Add an alarm for the chosen metric to be notified when the metric crosses a threshold you specify. For more information, see [Alarms](#) and [Create bucket metric alarms](#).

Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
- [Configuring bucket access permissions in Amazon Lightsail](#)
- [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
- [Creating access keys for a bucket in Amazon Lightsail](#)
- [Configuring resource access for a bucket in Amazon Lightsail](#)
- [Configuring cross-account access for a bucket in Amazon Lightsail](#)

5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).
7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).

14 Learn how to connect your bucket to other resources. For more information, see the following tutorials.

- [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
- [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)

15 Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Topics

- [Create Lightsail bucket metric alarms](#)

Create Lightsail bucket metric alarms

You can create an Amazon Lightsail alarm that watches a single bucket metric. An alarm can be configured to notify you based on the value of the metric relative to a threshold that you specify. Notifications can be a banner displayed in the Lightsail console, an email sent to your email address, and an SMS text message sent to your mobile phone number. For more information about alarms, see [Alarms](#).

Contents

- [Bucket alarm limits](#)
- [Best practices for configuring bucket alarms](#)
- [Default alarm settings](#)
- [Create bucket metric alarms using the Lightsail console](#)
- [Test bucket metric alarms using the Lightsail console](#)
- [Next steps after creating bucket alarms](#)

Bucket alarm limits

The following limits apply to alarms:

- You can configure two alarms per metric.
- Alarms are evaluated in 5 minute intervals, and each data point for alarms represents a 5 minute period of aggregated metric data.

- You can only configure an alarm to notify you when the alarm state changes to OK if you configure the alarm to notify you by email and/or SMS text message.
- You can only test the OK alarm notification if you configure the alarm to notify you by email and/or SMS text message.
- You can only configure an alarm to notify you when the alarm state changes to `INSUFFICIENT_DATA` if you configure the alarm to notify you by email and/or SMS text message, and if you choose the **Do not evaluate the missing data** option for missing data points.
- You can only test notifications if the alarm is in an OK state.

Best practices for configuring bucket alarms

Before you configure a metric alarm for your bucket, you should determine what you want to be notified about. For example, with the **Bucket size** metric in mind, you might want to be notified when your bucket is almost full. If your bucket current plan includes a 5 GB of storage space, then you might want to configure an alarm for the **Bucket size** metric when it reaches 4.5 GB. Then you should be notified with sufficient time to upsize your bucket's plan.


Default alarm settings

Default alarm settings are pre-populated when you add a new alarm in the Lightsail console. That is the recommended alarm configuration for the metric you selected. However, you should confirm that the default alarm configuration is appropriate for your resource. For example, the default alarm threshold for the bucket size bytes metric is **greater than or equal to 75 GB**. However, that request threshold might be too high for your bucket if it's configured to have only 5 GB of storage space. You might want to modify the alarm threshold to be **equal to or greater than 4.5 GB**.

Create bucket metric alarms using the Lightsail console

Complete the following steps to create a bucket metric alarm using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to create alarms.
4. Choose the **Metrics** tab on the bucket management page.
5. Choose the metric for which you want to create an alarm in the drop-down menu under the **Metrics Graphs** heading. For more information, see [Resource metrics](#).

6. Choose **Add alarm** in the **Alarms** section of the page.
 7. Choose a comparison operator value in the drop-down menu. Example values are greater than or equal to, greater than, less than, or less than or equal to.
 8. Enter a threshold for the alarm.
 9. Enter the data points to alarm.
 10. Choose the evaluation periods. The period can be specified in 5-minute increments, from 5 minutes up to 24 hours.
 11. Choose one of the following notification methods:
 - **Email** — You are notified by email when the alarm state changes to ALARM.
 - **SMS text message** — You are notified by SMS text message when the alarm state changes to ALARM. SMS messaging is not supported in all AWS Regions, and SMS text messages cannot be sent to all countries/regions. For more information, see [SMS text messaging support](#).
-  **Note**

You are required to add an email address or mobile phone number if you select to be notified by email or SMS but you haven't yet configured a notification contact in the resource's AWS Region. For more information, see [Notifications](#).
12. (Optional) Choose **Send me a notification when the alarm state change to OK** to be notified when the alarm state changes to OK. This option is available only if you choose to be notified by Email or SMS text message.
 13. (Optional) Choose **Advanced settings**, and then choose one of the following options:
 - Choose how the alarm should treat missing data The following options are available:
 - **Assume it's not within the threshold (Breaching threshold)** — Missing data points are treated as "bad" and breaching the threshold.
 - **Assume it's within the threshold (Not breaching threshold)** — Missing data points are treated as "good" and within the threshold.
 - **Use the value of the last good data point (Ignore and maintain the current alarm state)** — The current alarm state is maintained.
 - **Do not evaluate it (Treat missing data as missing)** — The alarm doesn't consider missing data points when evaluating whether to change state.

- Choose **Send a notification if there is insufficient data** to be notified when the alarm state changes to `INSUFFICIENT_DATA`. This option is available only if you choose to be notified by Email or SMS text message.

14. Choose **Create** to add the alarm.

To edit the alarm later, choose the ellipsis icon (:) next to the alarm you want to edit, and choose **Edit alarm**.

Test bucket metric alarms using the Lightsail console

Complete the following steps to test an alarm using the Lightsail console. You might want to test an alarm to confirm that the configured notification options are working, such as to ensure that you receive an email or an SMS text message when the alarm is triggered.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket for which you want to test an alarm.
4. Choose the **Metrics** tab on the bucket management page.
5. Choose the metric for which you want to test an alarm in the drop-down menu under the **Metrics Graphs** heading.
6. Scroll down to the **Alarms** section of the page, and choose the ellipsis icon (:) next to the alarm you want to test.
7. Choose one of the following options:
 - **Test alarm notification** — Choose this option to test the notifications for when the alarm state changes to `ALARM`.
 - **Test OK notification** — Choose this option to test the notifications for when the alarm state changes to `OK`.

Note

If either of these options is unavailable, you might not have configured the notification options for the alarm, or the alarm might currently be in an `ALARM` state. For more information, see [Bucket alarm limits](#).

The alarm momentarily changes to an ALARM or OK state depending on the test option you chose, and an email and/or SMS text message is sent depending on what you configured as the notification method for the alarm. A notification banner displays in the Lightsail console only if you chose to test the ALARM notification. A notification banner is not displayed if you chose to test the OK notification. The alarm will return to its actual state often after a few seconds.

Next steps after creating bucket alarms

There are a few additional tasks that you can perform for your bucket alarms:

- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete notification contacts](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

View Lightsail container service metrics

After you create an Amazon Lightsail container service, you can view its metric graphs on the Metrics tab of the service's management page. Monitoring metrics is an important part of maintaining the reliability, availability, and performance of your resources. Monitor and collect metric data from your resources regularly so that you can more readily debug a multi-point failure, if one occurs. For more information about metrics, see [Metrics in Amazon Lightsail](#).

When monitoring your resources, you should establish a baseline for normal resource performance in your environment.

Note

Alarms and notifications are currently not supported for container service metrics.

Container service metrics

The following container service metrics are available:

- **CPU utilization** — The average percentage of compute units that are currently in use across all nodes of your container service. This metric identifies the processing power required to run containers on your container service.
- **Memory utilization** — The average percentage of memory that is currently in use across all nodes of your container service. This metric identifies the memory required to run containers on your container service.

Note

If you create a new deployment, then the existing utilization metrics of your container service will disappear, and only metrics for the new current deployment will be shown.

View container service metrics in the Lightsail console

Complete the following procedure to view container service metrics in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Containers** tab.
3. Choose the name of the container for which you want to view metrics.
4. Choose the **Metrics** tab on the container service management page.
5. Choose the metric that you want to view in the dropdown menu under the **Metrics** graphs heading.

The graph displays a visual representation of the data points for the chosen metric.

6. You can perform the following actions on the metrics graph:
 - Change the view of the graph to show data for 1 hour, 6 hours, 1 day, 1 week, and 2 weeks.
 - Pause your cursor on a data point to view detailed information about that data point.

Note

Alarms and notifications are currently not supported for container service metrics.

View Lightsail database metrics

After you launch a database in Amazon Lightsail, you can view its metric graphs on the **Metrics** tab of the database's management page. Monitoring metrics is an important part of maintaining the reliability, availability, and performance of your resources. Monitor and collect metric data from your resources regularly so that you can more readily debug a multi-point failure, if one occurs. For more information about metrics, see [Metrics](#).

When monitoring your resources, you should establish a baseline for normal resource performance in your environment. After you've established a baseline, you can configure alarms in the Lightsail console to notify you when your resources are performing outside of specified thresholds. For more information, see [Notifications](#) and [Alarms](#).

Contents

- [Database metrics](#)
- [View database metrics](#)
- [Next steps after viewing your database metrics](#)

Database metrics

The following database metrics are available:

- **CPU utilization (CPUUtilization)** — The percentage of CPU utilization currently in use on the database.
- **Database connections (DatabaseConnections)** — The number of database connections in use.
- **Disk queue depth (DiskQueueDepth)** — The number of outstanding IOs (read/write requests) that are waiting to access the disk.
- **Free storage space (FreeStorageSpace)** — The amount of available storage space.
- **Network receive throughput (NetworkReceiveThroughput)** — The incoming (Receive) network traffic on the database, including both customer database traffic and AWS traffic used for monitoring and replication.
- **Network transmit throughput (NetworkTransmitThroughput)** — The outgoing (Transmit) network traffic on the database, including both customer database traffic and AWS traffic used for monitoring and replication.

Viewing database metrics in the Lightsail console

Complete the following steps to view database metrics in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database for which you want to view metrics.
4. Choose the **Metrics** tab on the database management page.
5. Choose the metric that you want to view in the drop-down menu under the **Metrics graphs** heading.

The graph displays a visual representation of the data points for the chosen metric.

6. You can perform the following actions on the metrics graph:
 - Change the view of the graph to show data for 1 hour, 6 hours, 1 day, 1 week, and 2 weeks.
 - Pause your cursor on a data point to view detailed information about that data point.
 - Add an alarm for the chosen metric to be notified when the metric crosses a threshold you specify. For more information, see [Alarms](#) and [Create database metric alarms](#).

Next steps after viewing your database metrics

There are a few additional tasks that you can perform for your database metrics:

- Add an alarm for the chosen metric to be notified when the metric crosses a threshold you specify. For more information, see [Alarms](#) and [Create database metric alarms](#).
- When an alarm is triggered, a notification banner is displayed in the Lightsail console. To be notified by email and SMS text message, you must add your email address and mobile phone number as notification contacts in each AWS Region where you want to monitor your resources. For more information, see [Adding notification contacts](#).
- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete or disable metric alarms](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

Topics

- [Create Lightsail database metric alarms](#)

Create Lightsail database metric alarms

You can create an Amazon Lightsail alarm that watches a single database metric. An alarm can be configured to notify you based on the value of the metric relative to a threshold that you specify. Notifications can be a banner displayed in the Lightsail console, an email sent to your email address, and an SMS text message sent to your mobile phone number. For more information about alarms, see [Alarms](#).

Contents

- [Database alarm limits](#)
- [Best practices for configuring database alarms](#)
- [Default alarm settings](#)
- [Create database metric alarms using the Lightsail console](#)
- [Test database metric alarms using the Lightsail console](#)
- [Next steps after creating database alarms](#)

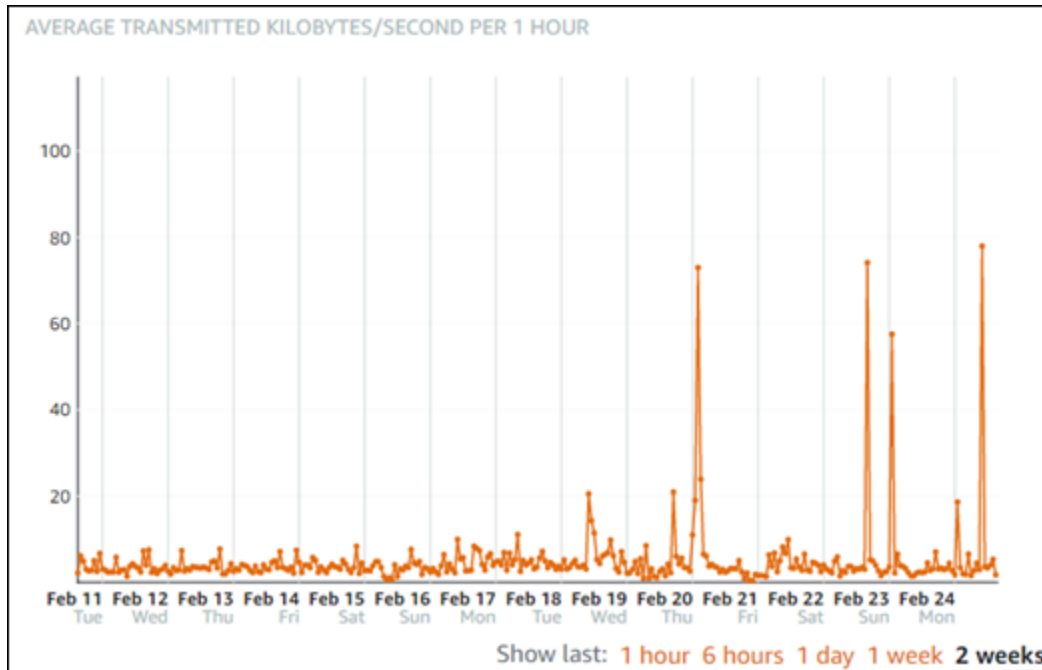
Database alarm limits

The following limits apply to alarms:

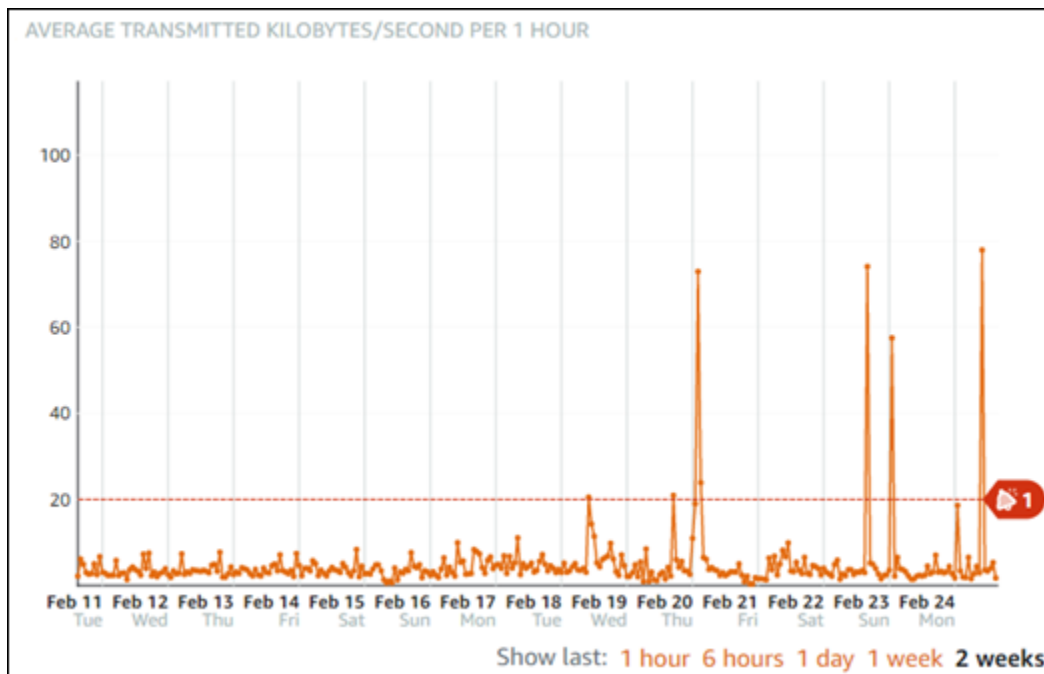
- You can configure two alarms per metric.
- Alarms are evaluated in 5 minute intervals, and each data point for alarms represents a 5 minute period of aggregated metric data.
- You can only configure an alarm to notify you when the alarm state changes to OK if you configure the alarm to notify you by email and/or SMS text message.
- You can only test the OK alarm notification if you configure the alarm to notify you by email and/or SMS text message.
- You can only configure an alarm to notify you when the alarm state changes to INSUFFICIENT_DATA if you configure the alarm to notify you by email and/or SMS text message, and if you choose the **Do not evaluate the missing data** option for missing data points.
- You can only test notifications if the alarm is in an OK state.

Best practices for configuring database alarms

Before you configure a metric alarm for your database, you should view the historical data of the metric. Identify the metric's low-levels, mid-levels, and high-levels over a period of the last two weeks. In the following network transmit throughput (`NetworkTransmitThroughput`) metric graph example, the low-levels are 0-10 KB/second per hour, the mid-levels are between 10-20 KB/second per hour, and the high-levels are between 20-80 KB/second per hour.



If you configure the alarm threshold to be **greater than or equal to** somewhere in the low-level range (e.g., 5 KB/second per hour), then you will get more frequent, and potentially unnecessary alarm notifications. If you configure the alarm threshold to be **greater than or equal to** somewhere in the high-level range (e.g., 20 KB per hour), then you will get less frequent alarm notifications, but that might be more important to investigate. When you configure an alarm, and enable it, an alarm line representing the threshold appears on the graph as shown in the following example. The alarm line labeled as 1 represents the threshold for Alarm 1, and the alarm line labeled as 2 represents the threshold for Alarm 2.



Default alarm settings


Default alarm settings are prepopulated when you add a new alarm in the Lightsail console. That is the recommended alarm configuration for the metric you selected. However, you should confirm that the default alarm configuration is appropriate for your resource. For example, the default alarm threshold for the free storage space (`FreeStorageSpace`) metric is **less than** 5 Bytes for 1 time within the last 5 minutes. However, that free storage space threshold might be too low for your database. You might want to modify the alarm threshold to be **less than** 4 GB for 1 time within the last 5 minutes.

Create database metric alarms using the Lightsail console

Complete the following steps to create a database metric alarm using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database for which you want to create alarms.
4. Choose the **Metrics** tab on the database management page.
5. Choose the metric for which you want to create an alarm in the drop-down menu under the **Metrics Graphs** heading. For more information, see [Resource metrics](#).
6. Choose **Add alarm** in the **Alarms** section of the page.

7. Choose a comparison operator value in the drop-down menu. Example values are greater than or equal to, greater than, less than, or less than or equal to.
8. Enter a threshold for the alarm.
9. Enter the data points to alarm.
10. Choose the evaluation periods. The period can be specified in 5-minute increments, from 5 minutes up to 24 hours.
11. Choose one of the following notification methods:
 - **Email** — You are notified by email when the alarm state changes to ALARM.
 - **SMS text message** — You are notified by SMS text message when the alarm state changes to ALARM. SMS messaging is not supported in all AWS Regions in which you can create Lightsail resources, and SMS text messages cannot be sent to all countries/regions. For more information, see [SMS text messaging support](#).

 **Note**

You are required to add an email address or mobile phone number if you select to be notified by email or SMS but you haven't yet configured a notification contact in the resource's AWS Region. For more information, see [Notifications](#).

12. (Optional) Choose **Send me a notification when the alarm state change to OK** to be notified when the alarm state changes to OK. This option is available only if you choose to be notified by Email or SMS text message.
13. (Optional) Choose **Advanced settings**, and then choose one of the following options:
 - Choose how the alarm should treat missing data. The following options are available:
 - **Assume it's not within the threshold (Breaching threshold)** — Missing data points are treated as "bad" and breaching the threshold.
 - **Assume it's within the threshold (Not breaching threshold)** — Missing data points are treated as "good" and within the threshold.
 - **Use the value of the last good data-point (Ignore and maintain the current alarm state)** — The current alarm state is maintained.
 - **Do not evaluate it (Treat missing data as missing)** — The alarm doesn't consider missing data points when evaluating whether to change state.

- Choose **Send a notification if there is insufficient data** to be notified when the alarm state changes to `INSUFFICIENT_DATA`. This option is available only if you choose to be notified by Email or SMS text message.

14. Choose **Create** to add the alarm.

To edit the alarm later, choose the ellipsis icon (:) next to the alarm you want to edit, and choose **Edit alarm**.

Testing database metric alarms using the Lightsail console

Complete the following steps to test an alarm using the Lightsail console. You might want to test an alarm to confirm that the configured notification options are working, such as to ensure that you receive an email or an SMS text message when the alarm is triggered.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Databases** tab.
3. Choose the name of the database for which you want to test an alarm.
4. Choose the **Metrics** tab on the database management page.
5. Choose the metric for which you want to test an alarm in the drop-down menu under the **Metrics Graphs** heading.
6. Scroll down to the **Alarms** section of the page, and choose the ellipsis icon (:) next to the alarm you want to test.
7. Choose one of the following options:
 - **Test alarm notification** — Choose this option to test the notifications for when the alarm state changes to `ALARM`.
 - **Test OK notification** — Choose this option to test the notifications for when the alarm state changes to `OK`.

Note

If either of these options is unavailable, you might not have configured the notification options for the alarm, or the alarm might currently be in an `ALARM` state. For more information, see [Database alarm limits](#).

The alarm momentarily changes to an ALARM or OK state depending on the test option you chose, and an email and/or SMS text message is sent depending on what you configured as the notification method for the alarm. A notification banner displays in the Lightsail console only if you chose to test the ALARM notification. A notification banner is not displayed if you chose to test the OK notification. The alarm will return to its actual state often after a few seconds.

Next steps after creating database alarms

There are a few additional tasks that you can perform for your database alarms:

- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete notification contacts](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

View Lightsail distribution metrics

After you create a distribution in Amazon Lightsail, you can view its metric graphs on the **Metrics** tab of the distribution's management page. Monitoring metrics is an important part of maintaining the reliability, availability, and performance of your resources. Monitor and collect metric data from your resources regularly so that you can more readily debug a multi-point failure, if one occurs. For more information about metrics, see [Metrics](#).

When monitoring your resources, you should establish a baseline for normal resource performance in your environment. Then you can configure alarms in the Lightsail console to notify you when your resources are performing outside of specified thresholds. For more information, see [Notifications](#) and [Alarms](#).

Contents

- [Distribution metrics](#)
- [View distribution metrics in the Lightsail console](#)
- [Next steps after viewing your distribution metrics](#)

Distribution metrics

The following distribution metrics are available:

- **Requests** — The total number of viewer requests received by your distribution, for all HTTP methods, and for both HTTP and HTTPS requests.
- **Bytes uploaded** — The number of bytes uploaded to your origin by your distribution, using POST and PUT requests.
- **Bytes downloaded** — The number of bytes downloaded by viewers for GET, HEAD, and OPTIONS requests.
- **Total error rate** — The percentage of all viewer requests for which the response's HTTP status code was 4xx or 5xx.
- **HTTP 4xx error rate** — The percentage of all viewer requests for which the response's HTTP status code was 4xx. In these cases, the client or client viewer may have made an error. For example, a status code of 404 (Not Found) means that the client requested an object that could not be found.
- **HTTP 5xx error rate** — The percentage of all viewer requests for which the response's HTTP status code was 5xx. In these cases, the origin server did not satisfy the request. For example, a status code of 503 (Service Unavailable) means that the origin server is currently unavailable.

View distribution metrics in the Lightsail console

Complete the following procedure to view distribution metrics in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which you want to view metrics.
4. Choose the **Metrics** tab on the distribution management page.
5. Choose the metric that you want to view in the drop-down menu under the **Metrics graphs** heading.

The graph displays a visual representation of the data points for the chosen metric.

6. You can perform the following actions on the metrics graph:
 - Change the view of the graph to show data for 1 hour, 6 hours, 1 day, 1 week, and 2 weeks.
 - Pause your cursor on a data point to view detailed information about that data point.

- Add an alarm for the chosen metric to be notified when the metric crosses a threshold you specify. For more information, see [Alarms](#) and [Create instance metric alarms](#).

Next steps after viewing your distribution metrics

There are a few additional tasks that you can perform for your distribution metrics:

- Add an alarm for the chosen metric to be notified when the metric crosses a threshold you specify. For more information, see [Alarms](#) and [Create distribution metric alarms](#).
- When an alarm is triggered, a notification banner is displayed in the Lightsail console. To be notified by email and SMS text message, you must add your email address and mobile phone number as notification contacts in each AWS Region where you want to monitor your resources. For more information, see [Add notification contacts](#).
- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete or disable metric alarms](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

Topics

- [Create Lightsail distribution metric alarms](#)

Create Lightsail distribution metric alarms

You can create an Amazon Lightsail alarm that watches a single distribution metric. An alarm can be configured to notify you based on the value of the metric relative to a threshold that you specify. Notifications can be a banner displayed in the Lightsail console, an email sent to your email address, and an SMS text message sent to your mobile phone number. For more information about alarms, see [Alarms](#).

Contents

- [Distribution alarm limits](#)
- [Best practices for configuring distribution alarms](#)
- [Default alarm settings](#)
- [Use the Lightsail console to create distribution metric alarms](#)

- [Test distribution metric alarms](#)
- [Next steps after creating distribution alarms](#)

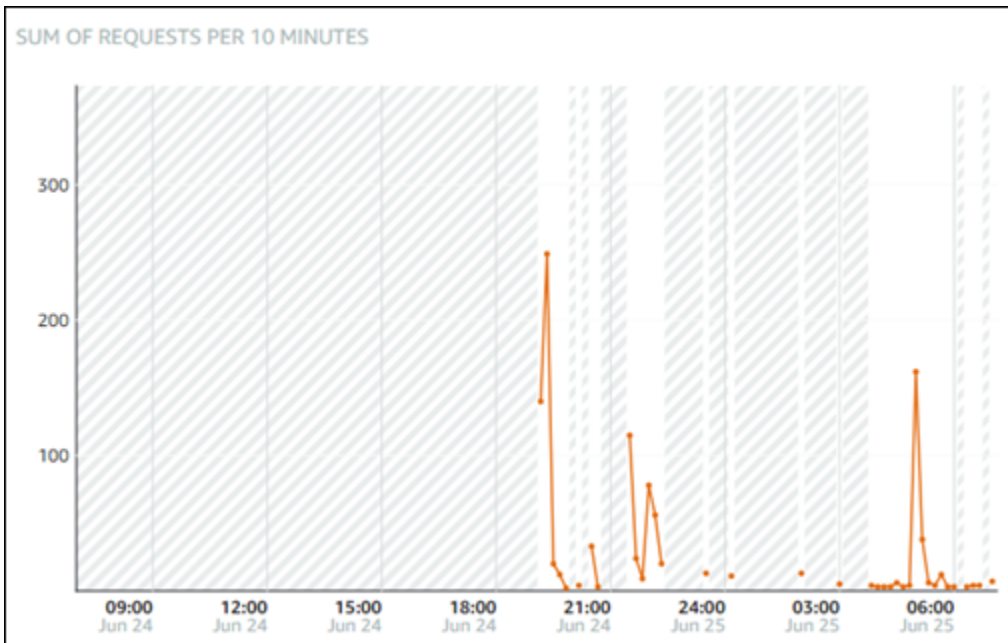
Distribution alarm limits

The following limits apply to alarms:

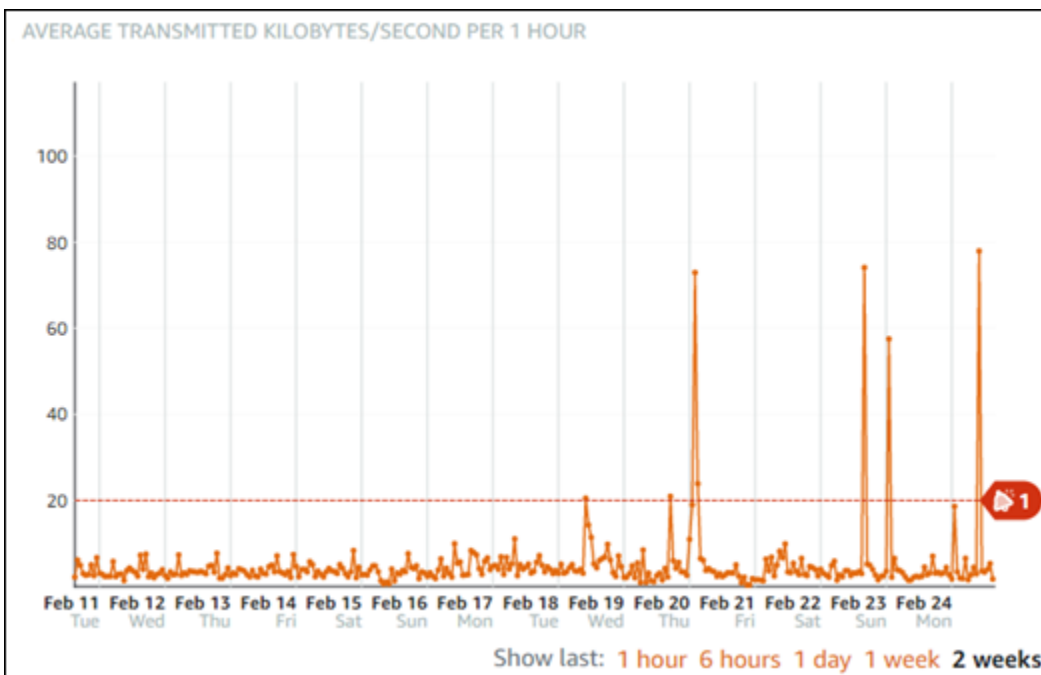
- You can configure two alarms per metric.
- Alarms are evaluated in 5 minute intervals, and each data point for alarms represents a 5 minute period of aggregated metric data.
- You can only configure an alarm to notify you when the alarm state changes to OK if you configure the alarm to notify you by email and/or SMS text message.
- You can only test the OK alarm notification if you configure the alarm to notify you by email and/or SMS text message.
- You can only configure an alarm to notify you when the alarm state changes to `INSUFFICIENT_DATA` if you configure the alarm to notify you by email and/or SMS text message, and if you choose the **Do not evaluate the missing data** option for missing data points.
- You can only test notifications if the alarm is in an OK state.

Best practices for configuring distribution alarms

Before you configure a metric alarm for your distribution, you should view the historical data of the metric. Identify the metric's low-levels, mid-levels, and high-levels over a period of the last two weeks. In the following requests metric graph example, the low-levels are 0-10 requests, the mid-levels are between 10-50 requests, and the high-levels are between 50-250 requests.



If you configure the alarm threshold to be **greater than or equal to** somewhere in the low-level range (e.g., 5 requests), then you will get more frequent, and potentially unnecessary alarm notifications. If you configure the alarm threshold to be **greater than or equal to** somewhere in the high-level range (e.g., 150 request), then you will get less frequent alarm notifications, but that might be more important to investigate. When you configure an alarm, and enable it, an alarm line representing the threshold appears on the graph as shown in the following example. The alarm line labeled as 1 represents the threshold for Alarm 1, and the alarm line labeled as 2 represents the threshold for Alarm 2.



Default alarm settings

Default alarm settings are prepopulated when you add a new alarm in the Lightsail console. That is the recommended alarm configuration for the metric you selected. However, you should confirm that the default alarm configuration is appropriate for your resource. For example, the default alarm threshold for the requests metric is **greater than** 45 requests for 3 times within the last 15 minutes. However, that request threshold might be too low for your distribution. You might want to modify the alarm threshold to be **greater than** 150 requests for 3 time within the last 15 minutes.

Use the Lightsail console to create distribution metric alarms

Complete the following steps to create a distribution metric alarm using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which you want to create alarms.
4. Choose the **Metrics** tab on the distribution management page.
5. Choose the metric for which you want to create an alarm in the drop-down menu under the **Metrics Graphs** heading. For more information, see [Resource metrics](#).
6. Choose **Add alarm** in the **Alarms** section of the page.
7. Choose a comparison operator value in the drop-down menu. Example values are greater than or equal to, greater than, less than, or less than or equal to.
8. Enter a threshold for the alarm.
9. Enter the data points to alarm.
10. Choose the evaluation periods. The period can be specified in 5-minute increments, from 5 minutes up to 24 hours.
11. Choose one of the following notification methods:
 - **Email** — You are notified by email when the alarm state changes to ALARM.
 - **SMS text message** — You are notified by SMS text message when the alarm state changes to ALARM. SMS messaging is not supported in all AWS Regions in which you can create Lightsail resources, and SMS text messages cannot be sent to all countries/regions. For more information, see [SMS text messaging support](#).

Note

You are required to add an email address or mobile phone number if you select to be notified by email or SMS but you haven't yet configured a notification contact in the resource's AWS Region. For more information, see [Notifications](#).

12. (Optional) Choose **Send me a notification when the alarm state change to OK** to be notified when the alarm state changes to OK. This option is available only if you choose to be notified by Email or SMS text message.
13. (Optional) Choose **Advanced settings**, and then choose one of the following options:
 - Choose how the alarm should treat missing data. The following options are available:
 - **Assume it's not within the threshold (Breaching threshold)** — Missing data points are treated as "bad" and breaching the threshold.
 - **Assume it's within the threshold (Not breaching threshold)** — Missing data points are treated as "good" and within the threshold.
 - **Use the value of the last good datapoint (Ignore and maintain the current alarm state)** — The current alarm state is maintained.
 - **Do not evaluate it (Treat missing data as missing)** — The alarm doesn't consider missing data points when evaluating whether to change state.
 - Choose **Send a notification if there is insufficient data** to be notified when the alarm state changes to INSUFFICIENT_DATA. This option is available only if you choose to be notified by Email or SMS text message.
14. Choose **Create** to add the alarm.


To edit the alarm later, choose the ellipsis icon (:) next to the alarm you want to edit, and choose **Edit alarm**.

Test distribution metric alarms

Complete the following steps to test an alarm using the Lightsail console. You might want to test an alarm to confirm that the configured notification options are working, such as to ensure that you receive an email or an SMS text message when the alarm is triggered.

1. Sign in to the [Lightsail console](#).

2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the distribution for which you want to test an alarm.
4. Choose the **Metrics** tab on the distribution management page.
5. Choose the metric for which you want to test an alarm in the drop-down menu under the **Metrics Graphs** heading.
6. Scroll down to the **Alarms** section of the page, and choose the ellipsis icon (:) next to the alarm you want to test.
7. Choose one of the following options:
 - **Test alarm notification** — Choose this option to test the notifications for when the alarm state changes to ALARM.
 - **Test OK notification** — Choose this option to test the notifications for when the alarm state changes to OK.

 **Note**

If either of these options is unavailable, you might not have configured the notification options for the alarm, or the alarm might currently be in an ALARM state. For more information, see [Distribution alarm limits](#).

The alarm momentarily changes to an ALARM or OK state depending on the test option you chose, and an email and/or SMS text message is sent depending on what you configured as the notification method for the alarm. A notification banner displays in the Lightsail console only if you chose to test the ALARM notification. A notification banner is not displayed if you chose to test the OK notification. The alarm will return to its actual state often after a few seconds.

Next steps after creating distribution alarms

There are a few additional tasks that you can perform for your distribution alarms:

- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete notification contacts](#). You can also disable or delete an alarm

to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

View Lightsail load balancer health metrics

After you create a load balancer in Amazon Lightsail, and attach instances to it, you can view its metric graphs on the Metrics tab of the load balancer's management page. Monitoring metrics is an important part of maintaining the reliability, availability, and performance of your resources. Monitor and collect metric data from your resources regularly so that you can more readily debug a multi-point failure, if one occurs. For more information about metrics, see [Metrics](#).

When monitoring your resources, you should establish a baseline for normal resource performance in your environment. After you've established a baseline, you can configure alarms in the Lightsail console to notify you when your resources are performing outside of specified thresholds. For more information, see [Notifications](#) and [Alarms](#).

Contents

- [Load balancer metrics](#)
- [View load balancer metrics](#)
- [Next steps](#)

Load balancer metrics

The following load balancer metrics are available:

- **Healthy host count (HealthyHostCount)** — The number of target instances that are considered healthy.
- **Unhealthy host count (UnhealthyHostCount)** — The number of target instances that are considered unhealthy.
- **Load balancer HTTP 4XX (HTTPCode_LB_4XX_Count)** — The number of HTTP 4XX client error codes that originated from the load balancer. Client errors are generated when requests are malformed or incomplete. These requests were not received by the target instance. This count does not include response codes generated by the target instances.
- **Load balancer HTTP 5XX (HTTPCode_LB_5XX_Count)** — The number of HTTP 5XX server error codes that originated from the load balancer. This does not include any response codes

generated by the target instance. This metric is reported if there are no healthy instances attached to the load balancer, or if the request rate exceeds the capacity of the instances (spillover) or the load balancer.

- **Instance HTTP 2XX (HTTPCode_Instance_2XX_Count)** — The number of HTTP 2XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance HTTP 3XX (HTTPCode_Instance_3XX_Count)** — The number of HTTP 3XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance HTTP 4XX (HTTPCode_Instance_4XX_Count)** — The number of HTTP 4XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance HTTP 5XX (HTTPCode_Instance_5XX_Count)** — The number of HTTP 5XX response codes generated by the target instances. This does not include any response codes generated by the load balancer.
- **Instance response time (InstanceResponseTime)** — The time elapsed, in seconds, after the request leaves the load balancer until a response from the target instance is received.
- **Client TLS negotiation error count (ClientTLSNegotiationErrorCount)** — The number of TLS connections initiated by the client that did not establish a session with the load balancer due to a TLS error generated by the load balancer. Possible causes include a mismatch of ciphers or protocols.
- **Request count (RequestCount)** — The number of requests processed over IPv4. This count includes only the requests with a response generated by a target instance of the load balancer.
- **Rejected connection count (RejectedConnectionCount)** — The number of connections that were rejected because the load balancer had reached its maximum number of connections.

View load balancer metrics

Complete the following steps to view load balancer metrics in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the load balancer for which you want to view metrics.
4. Choose the **Metrics** tab on the load balancer management page.

5. Choose the metric that you want to view in the drop-down menu under the **Metrics graphs** heading.

The graph displays a visual representation of the data points for the chosen metric.

6. You can perform the following actions on the metrics graph:
 - Change the view of the graph to show data for 1 hour, 6 hours, 1 day, 1 week, and 2 weeks.
 - Pause your cursor on a data point to view detailed information about that data point.
 - Add an alarm for the chosen metric to be notified when the metric crosses a threshold you specify. For more information, see [Alarms](#) and [Create load balancer metric alarms](#).

Next steps

There are a few additional tasks that you can perform for your load balancer metrics:

- Add an alarm for the chosen metric to be notified when the metric crosses a threshold you specify. For more information, see [Alarms](#) and [Create load balancer metric alarms](#).
- When an alarm is triggered, a notification banner is displayed in the Lightsail console. To be notified by email and SMS text message, you must add your email address and mobile phone number as notification contacts in each AWS Region where you want to monitor your resources. For more information, see [Add notification contacts](#).
- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete or disable metric alarms](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

Topics

- [Create Lightsail load balancer metric alarms](#)

Create Lightsail load balancer metric alarms

You can create an Amazon Lightsail alarm that watches a single load balancer metric. An alarm can be configured to notify you based on the value of the metric relative to a threshold that you specify. Notifications can be a banner displayed in the Lightsail console, an email sent to your email address, and an SMS text message sent to your mobile phone number. For more information about alarms, see [Alarms](#).

Contents

- [Load balancer alarm limits](#)
- [Best practices for configuring load balancer alarms](#)
- [Default alarm settings](#)
- [Create load balancer metric alarms using the Lightsail console](#)
- [Test load balancer metric alarms using the Lightsail console](#)
- [Next steps](#)

Load balancer alarm limits

The following limits apply to alarms:

- You can configure two alarms per metric.
- Alarms are evaluated in 5 minute intervals, and each data point for alarms represents a 5 minute period of aggregated metric data.
- You can only configure an alarm to notify you when the alarm state changes to OK if you configure the alarm to notify you by email and/or SMS text message.
- You can only test the OK alarm notification if you configure the alarm to notify you by email and/or SMS text message.
- You can only configure an alarm to notify you when the alarm state changes to INSUFFICIENT_DATA if you configure the alarm to notify you by email and/or SMS text message, and if you choose the **Do not evaluate the missing data** option for missing data points.
- You can only test notifications if the alarm is in an OK state.

Best practices for configuring load balancer alarms

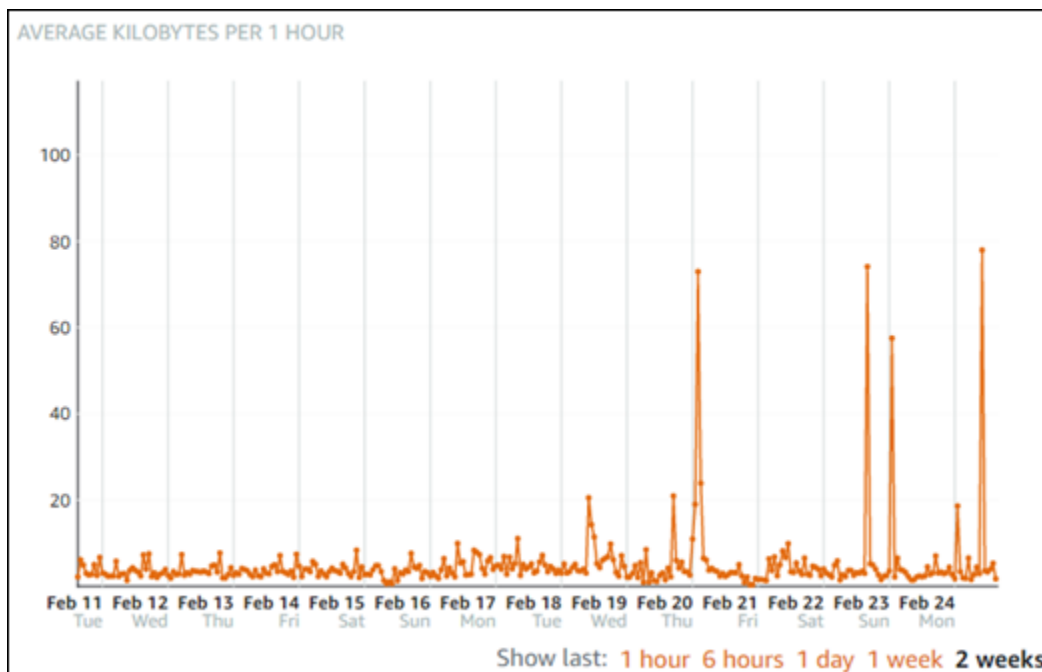
The following limits apply to alarms:

- You can configure two alarms per metric.
- Alarms are evaluated in 5 minute intervals, and each data point for alarms represents a 5 minute period of aggregated metric data.
- You can only configure an alarm to notify you when the alarm state changes to OK if you configure the alarm to notify you by email and/or SMS text message.

- You can only test the OK alarm notification if you configure the alarm to notify you by email and/or SMS text message.
- You can only configure an alarm to notify you when the alarm state changes to `INSUFFICIENT_DATA` if you configure the alarm to notify you by email and/or SMS text message, and if you choose the **Do not evaluate the missing data** option for missing data points.
- You can only test notifications if the alarm is in an OK state.

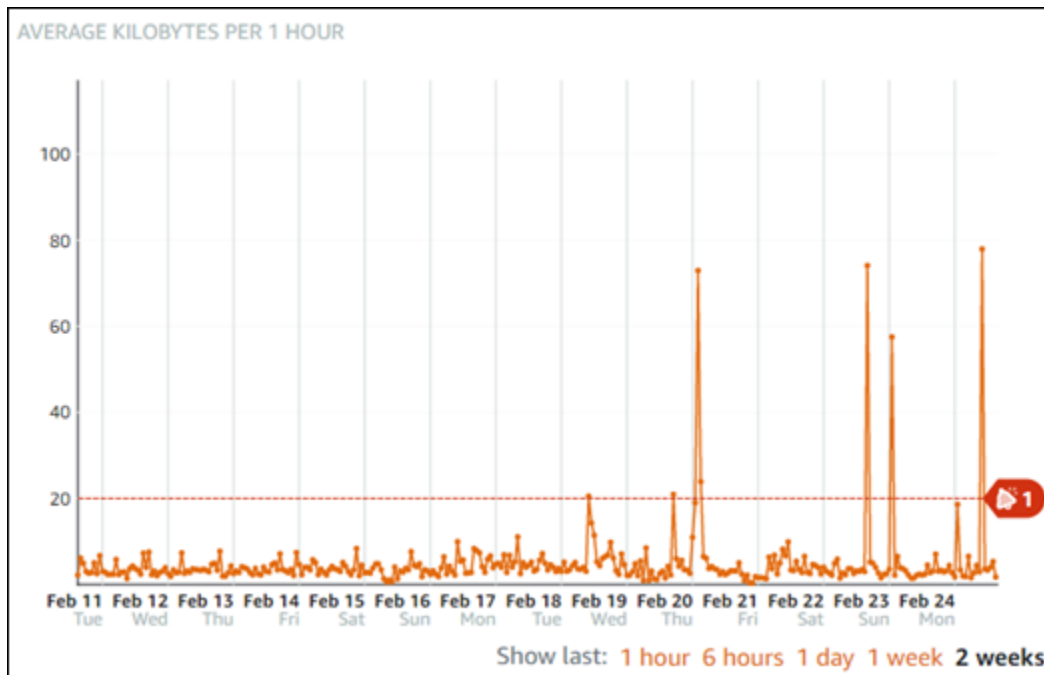
Default alarm settings

Before you configure a metric alarm, you should view the historical data of the metric. Identify the metric's low-levels, mid-levels, and high-levels over a period of the last two weeks. In the following instance outgoing network traffic (`NetworkOut`) metric graph example, the low-levels are 0-10 KB per hour, the mid-levels are between 10-20 KB per hour, and the high-levels are between 20-80 KB per hour.



If you configure the alarm threshold to be **greater than or equal to** somewhere in the low-level range (e.g., 5 KB per hour), then you will get more frequent, and potentially unnecessary alarm notifications. If you configure the alarm threshold to be **greater than or equal to** somewhere in the high-level range (e.g., 20 KB per hour), then you will get less frequent alarm notifications, but that might be more important to investigate. When you configure an alarm, and enable it, an alarm line representing the threshold appears on the graph as shown in the following example. The alarm line

labeled as 1 represents the threshold for Alarm 1, and the alarm line labeled as 2 represents the threshold for Alarm 2.




Create load balancer metric alarms using the Lightsail console

Complete the following steps to create a load balancer metric alarm using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the load balancer for which you want to create alarms.
4. Choose the **Metrics** tab on the load balancer management page.
5. Choose the metric for which you want to create an alarm in the drop-down menu under the **Metrics Graphs** heading. For more information, see [Resource metrics](#).
6. Choose **Add alarm** in the **Alarms** section of the page.
7. Choose a comparison operator value in the drop-down menu. Example values are greater than or equal to, greater than, less than, or less than or equal to.
8. Enter a threshold for the alarm.
9. Enter the data points to alarm.
10. Choose the evaluation periods. The period can be specified in 5-minute increments, from 5 minutes up to 24 hours.
11. Choose one of the following notification methods:

- **Email** — You are notified by email when the alarm state changes to ALARM.
- **SMS text message** — You are notified by SMS text message when the alarm state changes to ALARM. SMS messaging is not supported in all AWS Regions in which you can create Lightsail resources, and SMS text messages cannot be sent to all countries/regions. For more information, see [SMS text messaging support](#).

 **Note**

You are required to add an email address or mobile phone number if you select to be notified by email or SMS but you haven't yet configured a notification contact in the resource's AWS Region. For more information, see [Notifications](#).

12. (Optional) Choose **Send me a notification when the alarm state change to OK** to be notified when the alarm state changes to OK. This option is available only if you choose to be notified by Email or SMS text message.
13. (Optional) Choose **Advanced settings**, and then choose one of the following options:
 - Choose how the alarm should treat missing data The following options are available:
 - **Assume it's not within the threshold (Breaching threshold)** — Missing data points are treated as "bad" and breaching the threshold.
 - **Assume it's within the threshold (Not breaching threshold)** — Missing data points are treated as "good" and within the threshold.
 - **Use the value of the last good data point (Ignore and maintain the current alarm state)** — The current alarm state is maintained.
 - **Do not evaluate it (Treat missing data as missing)** — The alarm doesn't consider missing data points when evaluating whether to change state.
 - Choose **Send a notification if there is insufficient data** to be notified when the alarm state changes to INSUFFICIENT_DATA. This option is available only if you choose to be notified by Email or SMS text message.
14. Choose **Create** to add the alarm.

To edit the alarm later, choose the ellipsis icon (:) next to the alarm you want to edit, and choose **Edit alarm**.

Test load balancer metric alarms using the Lightsail console

Complete the following steps to test an alarm using the Lightsail console. You might want to test an alarm to confirm that the configured notification options are working, such as to ensure that you receive an email or an SMS text message when the alarm is triggered.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose the name of the load balancer for which you want to test an alarm.
4. Choose the **Metrics** tab on the load balancer management page.
5. Choose the metric for which you want to test an alarm in the drop-down menu under the **Metrics Graphs** heading.
6. Scroll down to the **Alarms** section of the page, and choose the ellipsis icon (:) next to the alarm you want to test.
7. Choose one of the following options:
 - **Test alarm notification** — Choose this option to test the notifications for when the alarm state changes to ALARM.
 - **Test OK notification** — Choose this option to test the notifications for when the alarm state changes to OK.

Note

If either of these options is unavailable, you might not have configured the notification options for the alarm, or the alarm might currently be in an ALARM state. For more information, see [Load balancer alarm limits](#).

The alarm momentarily changes to an ALARM or OK state depending on the test option you chose, and an email and/or SMS text message is sent depending on what you configured as the notification method for the alarm. A notification banner displays in the Lightsail console only if you chose to test the ALARM notification. A notification banner is not displayed if you chose to test the OK notification. The alarm will return to its actual state often after a few seconds.

Next steps after creating load balancer alarms

There are a few additional tasks that you can perform for your load balancer alarms:

- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete notification contacts](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

Add notification contacts in Lightsail

You can configure Amazon Lightsail to notify you when a metric for one of your instances, databases, load balancers, or content delivery network (CDN) distributions crosses a specified threshold. Notifications can be in the form of a banner displayed in the Lightsail console, an email sent to an address you specify, or an SMS text message sent to a mobile phone number you specify. To be notified by email and SMS text message, you must add your email address and mobile phone number as notification contacts in each AWS Region where you want to monitor your resources. For more information about notifications, see [Notifications](#).

Important

SMS text messaging feature has been temporarily disabled and is currently not supported in any AWS Region in which you can create Lightsail resources. For more information, see [SMS text messaging support](#).

Contents

- [Regional notification contact limits](#)
- [SMS text messaging support](#)
- [Email contact verification](#)
- [Adding notification contacts using the Lightsail console](#)
- [Adding notification contacts using the AWS CLI](#)
- [Next steps after adding your notification contacts](#)

Regional notification contact limits

You can add only one email address and one mobile phone number in each AWS Region. If you add an email address or mobile phone number in a Region where those were already added, you will be asked if you would like to replace the existing notification contact with the new contact.

If you require multiple email recipients in an AWS Region, you can configure a distribution list that forwards to multiple recipients, and add the distribution list's email address as the notification contact.

SMS text messaging support

Important

SMS text messaging feature has been temporarily disabled and is currently not supported in any AWS Region in which you can create Lightsail resources. Alternatively, you can configure email messaging or rely on the notification banners displayed in the Lightsail console.

The following information for SMS text messaging support is published for customers who configured SMS text messaging before we disabled the feature.

SMS text messaging is not supported in all AWS Regions in which you can create Lightsail resources. Also, SMS text messages cannot be sent to some countries and regions of the world. For AWS Regions in which SMS messaging is not supported, you can configure only an email notification contact.

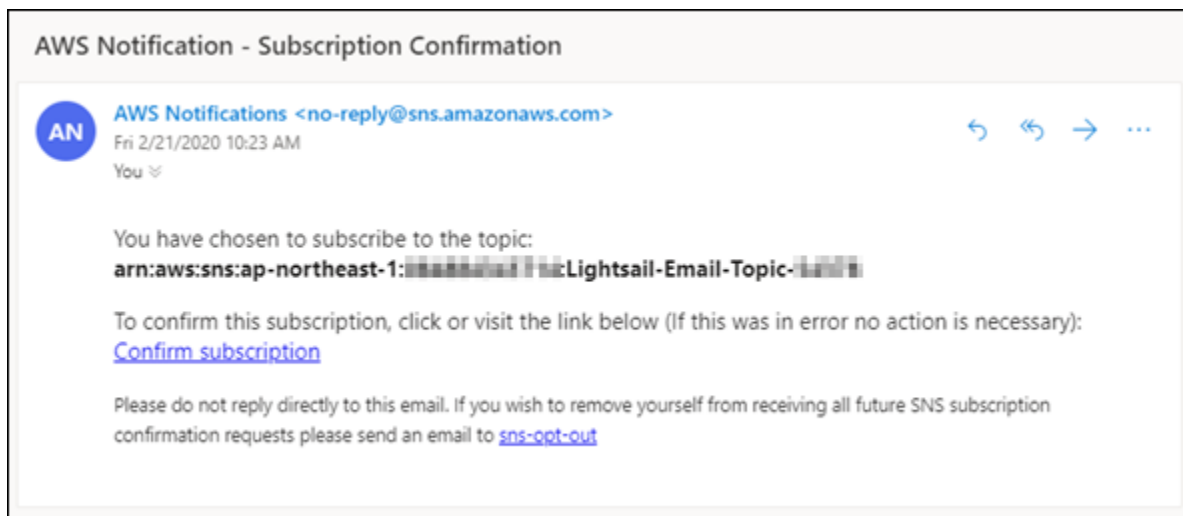
SMS messaging is supported in the following AWS Regions. These are Regions where SMS text messaging is supported by the Amazon Simple Notification Service (Amazon SNS), which is used by Lightsail to send you notifications:

- US East (N. Virginia) (us-east-1)
- US West (Oregon) (us-west-2)
- Asia Pacific (Singapore) (ap-southeast-1)
- Asia Pacific (Sydney) (ap-southeast-2)
- Asia Pacific (Tokyo) (ap-northeast-1)
- Europe (Ireland) (eu-west-1)

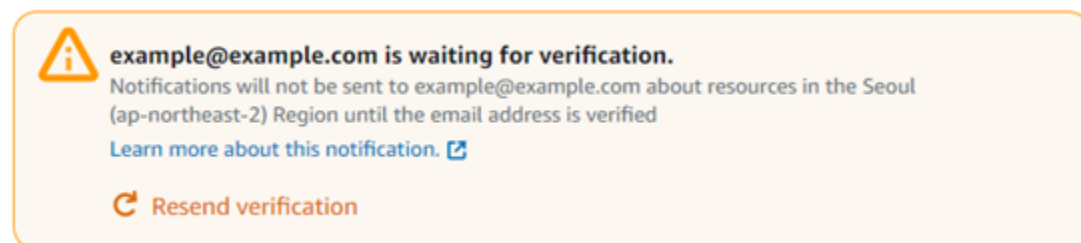
For a list of countries and regions of the world where SMS text messages can be sent, and the latest AWS Regions in which SMS text messaging is supported, see [Supported Regions and Countries](#) in the *Amazon SNS Developer Guide*.

Email contact verification

When you add an email address as a notification contact in Lightsail, a verification request is sent to that address. The verification request email contains a link that the recipient must click to confirm that they want to receive Lightsail notifications. Notifications are not sent to the email address until after it is verified. The verification comes from *AWS Notifications <no-reply@sns.amazonaws.com>*, with a subject of *AWS Notification - Subscription Confirmation*. SMS messaging does not require verification.



Check the mailbox's spam and junk folders if the verification request is not in the inbox folder. If the verification request got lost, or was deleted, choose **Resend verification** in the notification banner that is displayed in the Lightsail console, and in the **Account** page.

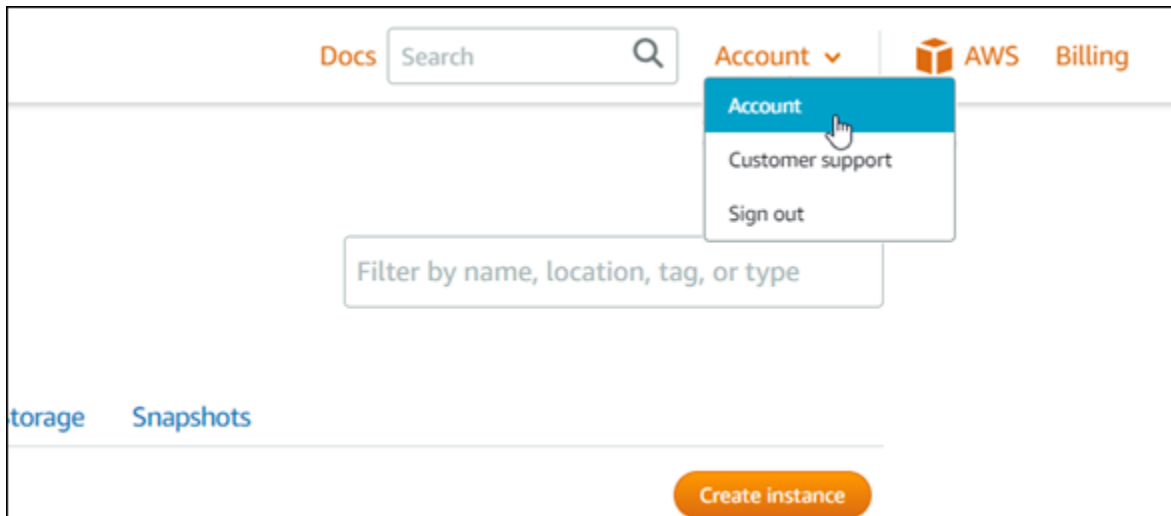


Adding notification contacts using the Lightsail console

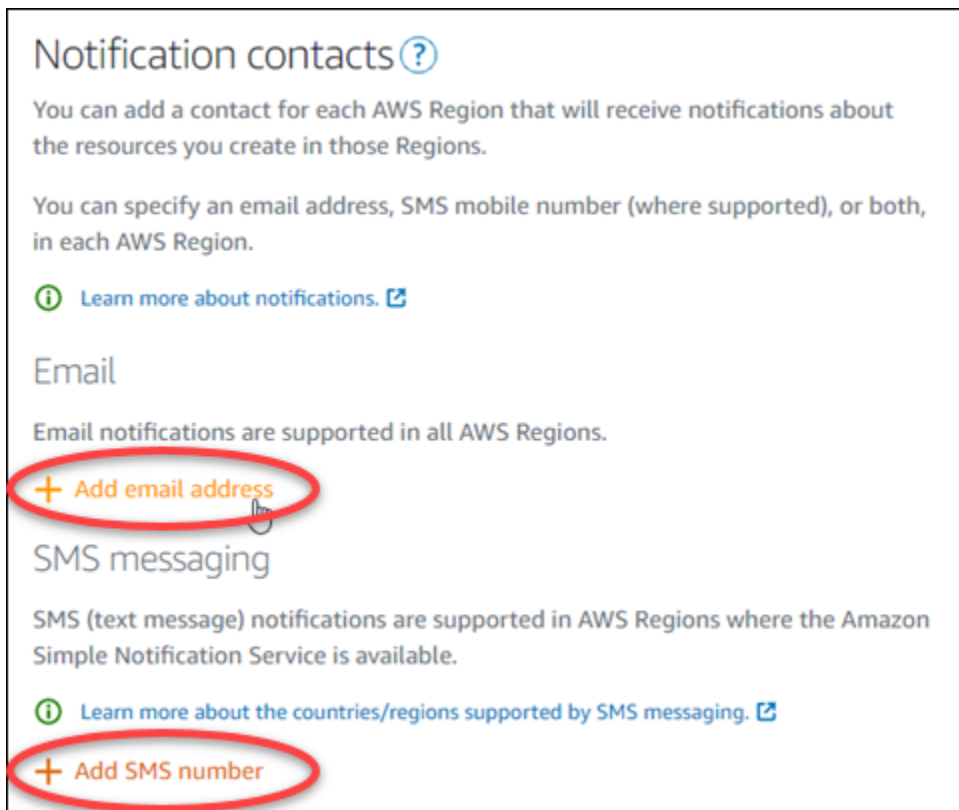
Complete the following steps to add notification contacts using the Lightsail console.

1. Sign in to the [Lightsail console](#).

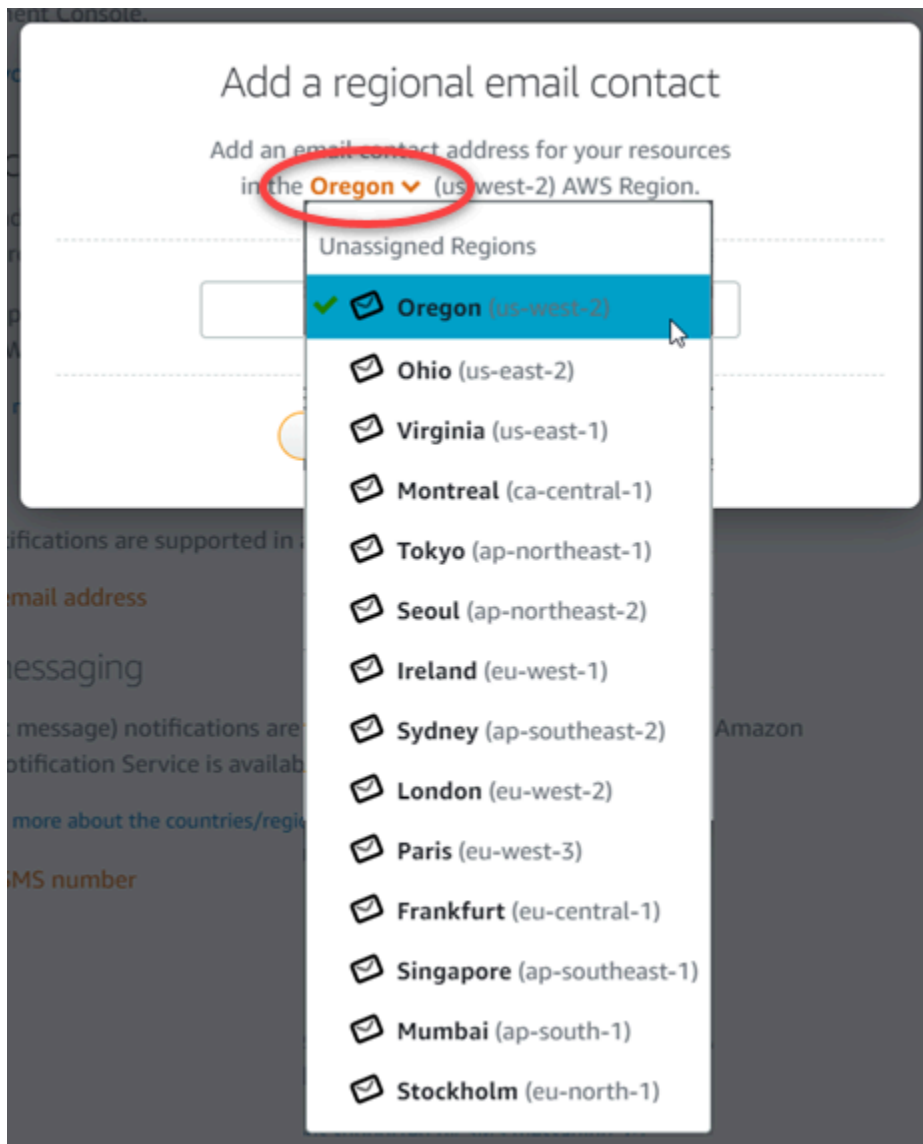
2. On the Lightsail home page, choose **Account** on the top navigation menu.
3. Choose **Account** in the drop-down menu.



4. Choose **Add email address** or **Add SMS number** in the **Notification contacts** section on the **Profile & contacts** tab.



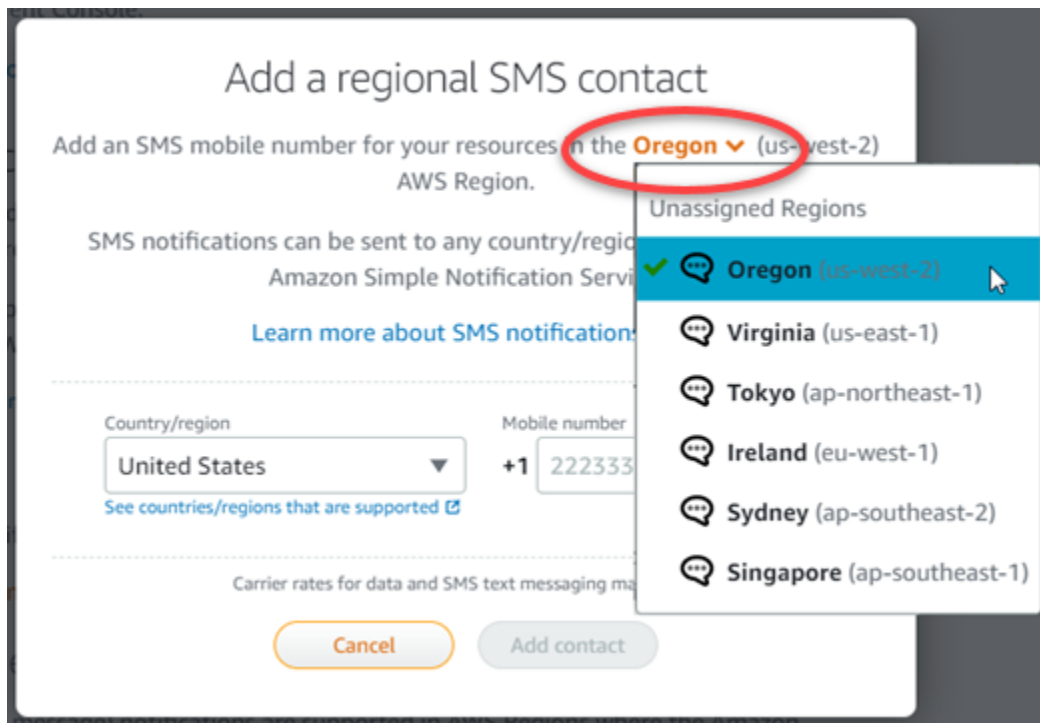
5. Complete one of the following steps:
 - If you are adding an email address, choose the AWS Region where you want to add the notification contact. Enter your email address into the text box.



- If you are adding an SMS number, choose the AWS Region where you want to add the notification contact. Choose the country of your mobile number, and enter it into the text box. The country code is already entered for you.

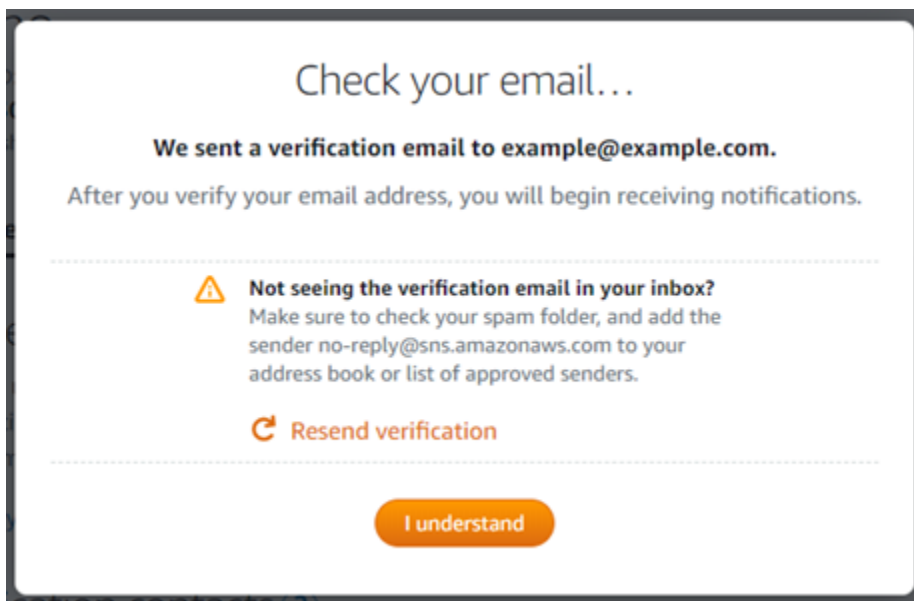
⚠ Important

SMS text messaging feature has been temporarily disabled and is currently not supported in any AWS Region in which you can create Lightsail resources. For more information, see [SMS text messaging support](#).



6. Choose **Add contact**.

When you add an email address as a notification contact, a verification request is sent to that address. The verification request email contains a link that the recipient must click to confirm that they want to receive Lightsail notifications. SMS messaging does not require verification.



7. Choose **I understand**.

Your email address or mobile phone number is added to the **Notification contacts** section. Email addresses are not verified until you complete the verification process in the following steps. Notifications are not sent to the email address until after you verify it. Choose **Resend** next to one of your regional email addresses to send another verification request if the verification request got lost, or was deleted.

Note

SMS messaging does not require verification. Therefore, you don't need to complete steps 8 through 10 in this procedure after you add an SMS notification contact.

The screenshot displays the 'Email' and 'SMS messaging' sections of the Lightsail console. The 'Email' section includes a table with one entry: 'example@example.com' in the 'Oregon (us-west-2)' region, which is not verified. A 'Resend' button and a trash icon are visible for this entry. The 'SMS messaging' section includes a table with one entry: '+1 222 333 4444' in the 'Oregon (us-west-2)' region. A trash icon is visible for this entry.

Email

Email notifications are supported in all AWS Regions.

[+ Add email address](#)

Email	Region	Verified	
example@example.com	Oregon (us-west-2)	No Resend	

SMS messaging

SMS (text message) notifications are supported in AWS Regions where the Amazon Simple Notification Service is available.

[Learn more about the countries/regions supported by SMS messaging.](#)

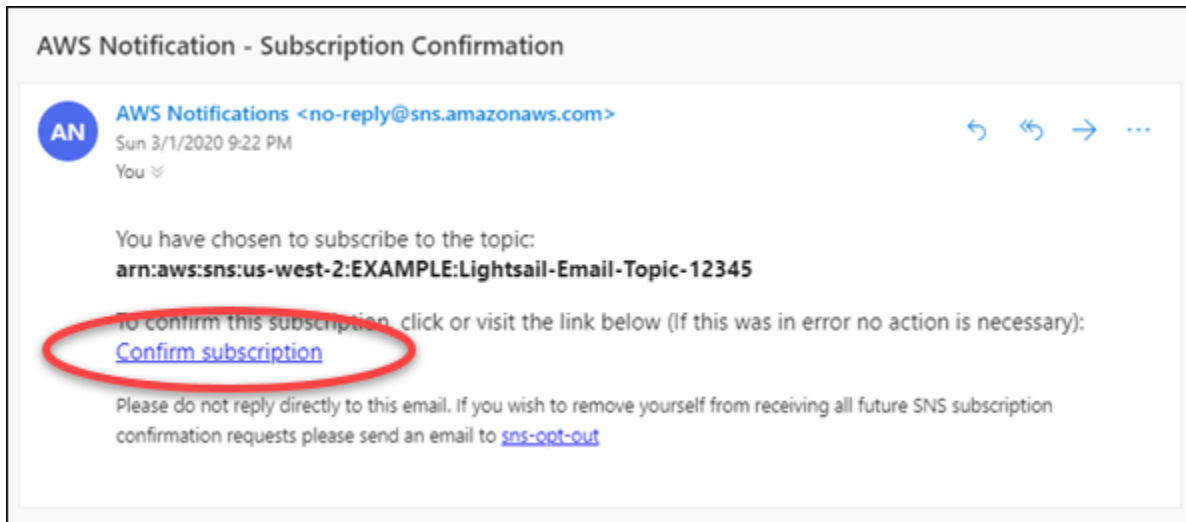
[+ Add SMS number](#)

Number	Region	
+1 222 333 4444	Oregon (us-west-2)	

8. Open the inbox for the email address that you added as a notification contact in Lightsail.
9. Open the **AWS Notification - Subscription Confirmation** email from **no-reply@sns.amazonaws.com**.

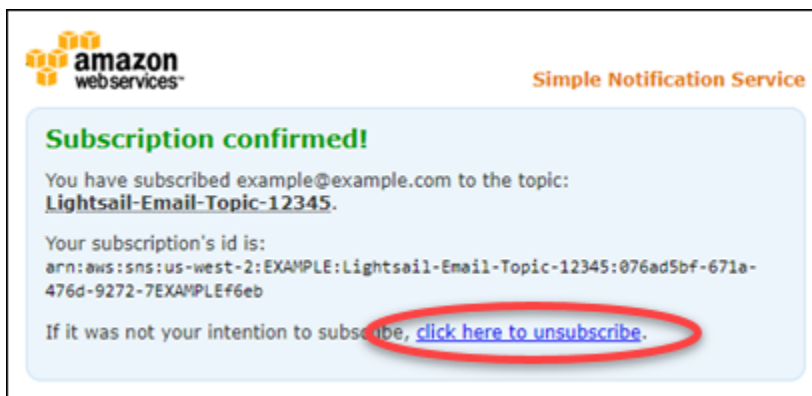
Note

Check the mailbox's spam and junk folders if the verification request is not in the inbox folder.



10. Choose **Confirm subscription** in the email to confirm that you want to receive Lightsail notifications.

A browser window opens to the following page confirming your subscription. To unsubscribe, choose **click here to unsubscribe** on the page. Or, if you have closed the page, complete the steps to [delete your notification contacts](#).



Adding notification contacts using the AWS CLI

Complete the following steps to add notification contacts for Lightsail using the AWS Command Line Interface (AWS CLI).

1. Open a Terminal or Command Prompt window.

If you haven't already, [install the AWS CLI](#) and [configure it to work with Lightsail](#).

2. Enter the following command to add a notification contact:

```
aws lightsail create-contact-method --region Region --notificationProtocol Protocol
--contact-endpoint Destination
```

In the command, replace:

- *Region* with the AWS Region in which the notification contact should be added.
- *Protocol* with the notification protocol for the contact, which should be Email or SMS.
- *Destination* with your email address or mobile phone number.

Note

Use the E.164 format when specifying a mobile phone number. E.164 is a standard for the phone number structure used for international telecommunication. Phone numbers that follow this format can have a maximum of 15 digits, and are prefixed with the plus character (+) and the country code. For example, a U.S. phone number in [E.164](#) format is specified as +1XXX5550100. For more information, see E.164 in Wikipedia.

Examples:

```
aws lightsail create-contact-method --region us-west-2 --notificationProtocol Email
--contact-endpoint example@example.com
```

```
aws lightsail create-contact-method --region us-east-1 --notificationProtocol SMS
--contact-endpoint +14445556666
```

When you press enter, you'll see an operation response with details about your request.

A verification request is sent to the email address that you specified as a notification contact. This confirms that the recipient wants to subscribe to Lightsail notifications. Email addresses are not verified until after the verification process in the following steps is completed. Notifications are not sent to the email address until after the email address is verified. Choose **Resend** next to one of your regional email addresses to send another verification request if the original notification is misplaced.

Note

SMS messaging does not require verification. Therefore, you don't need to complete steps 8 through 10 in this procedure when you add an SMS notification contact.

3. Open the inbox for the email address that you added as a notification contact.
4. Open the **AWS Notification - Subscription Confirmation** email from **no-reply@sns.amazonaws.com**.
5. Choose **Confirm subscription** in the email to confirm that you want to receive email notifications from Lightsail.

A browser window opens to the following page confirming your subscription. To unsubscribe, choose **click here to unsubscribe** on the page. Or, if you have closed the page, complete the steps to [delete your notification contacts](#).

Next steps after adding your notification contacts

There are a couple of additional tasks that you can perform for your notification contacts:

- Add an alarm in the AWS Region where you added your notification contacts. You can choose to be notified by email and SMS text message when the alarm starts. For more information, see [Alarms](#).
- If don't receive notifications when you expect to be notified, then there are a few things you should check to confirm that your notification contacts are configured correctly. To learn more, see [Troubleshooting Notifications](#).
- To stop receiving notifications, you can remove your email and mobile phone from Lightsail. For more information, see [Delete or disable metric alarms](#). You can also disable or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

Delete Lightsail notification contacts

Delete your email and mobile phone number notification contacts from Amazon Lightsail to stop receiving email and SMS text message notifications for your Lightsail resources. For more information about notifications, see [Notifications](#).

You can also disable, or delete an alarm to stop receiving notifications for a specific alarm. For more information, see [Delete or disable metric alarms](#).

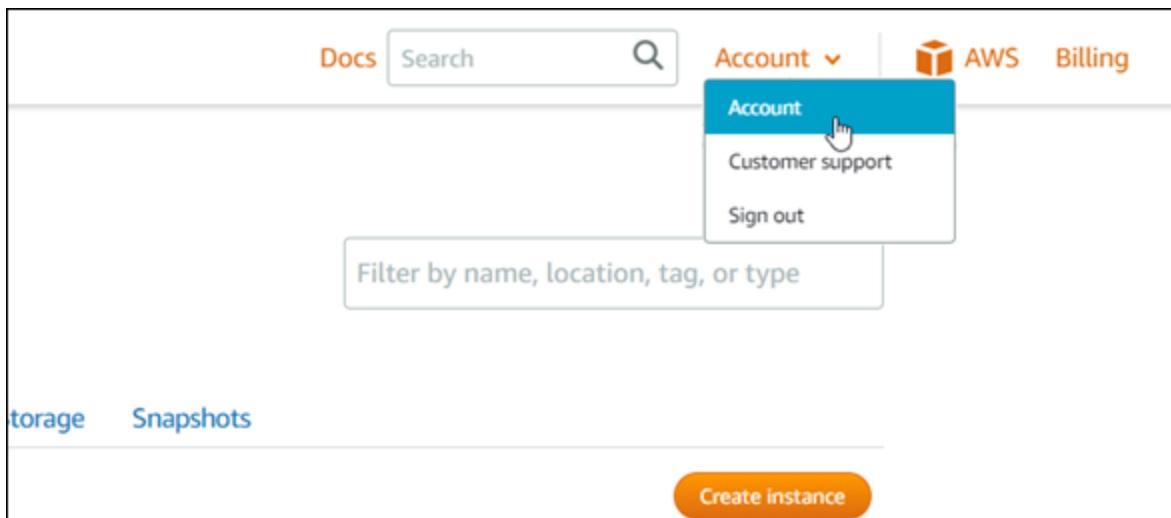
Contents

- [Deleting notification contacts using the Lightsail console](#)
- [Deleting notification contacts using the AWS CLI](#)
- [Next steps after deleting your notification contacts](#)

Deleting notification contacts using the Lightsail console

Complete the following steps to delete notification contacts using the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose **Account** on the top navigation menu.
3. Choose **Account** in the drop-down menu.



4. Choose the delete icon next to the email address or mobile phone number that you want to delete in the **Notification contacts** section on the **Profile & contacts** tab.
5. Choose **Yes** to confirm that you want to delete the notification contact.

Deleting notification contacts using the AWS CLI

Complete the following steps to delete notification contacts for Lightsail using the AWS Command Line Interface (AWS CLI).

1. Open a Terminal or Command Prompt window.

If you haven't already, [install the AWS CLI](#) and [configure it to work with Lightsail](#).

2. Enter the following command to delete a notification contact:

```
aws lightsail delete-contact-method --region Region --notificationProtocol Protocol
```

In the command, replace:

- *Region* with the AWS Region in which the notification contact should be deleted.
- *Protocol* with the notification protocol for the contact that you want to delete, such as Email or SMS.

Example:

```
aws lightsail delete-contact-method --region us-west-2 --notificationProtocol SMS
```

When you press enter, you'll see an operation response with details about your request.

Next steps after deleting your notification contacts

There are a couple of additional tasks that you can perform after deleting your notification contacts:

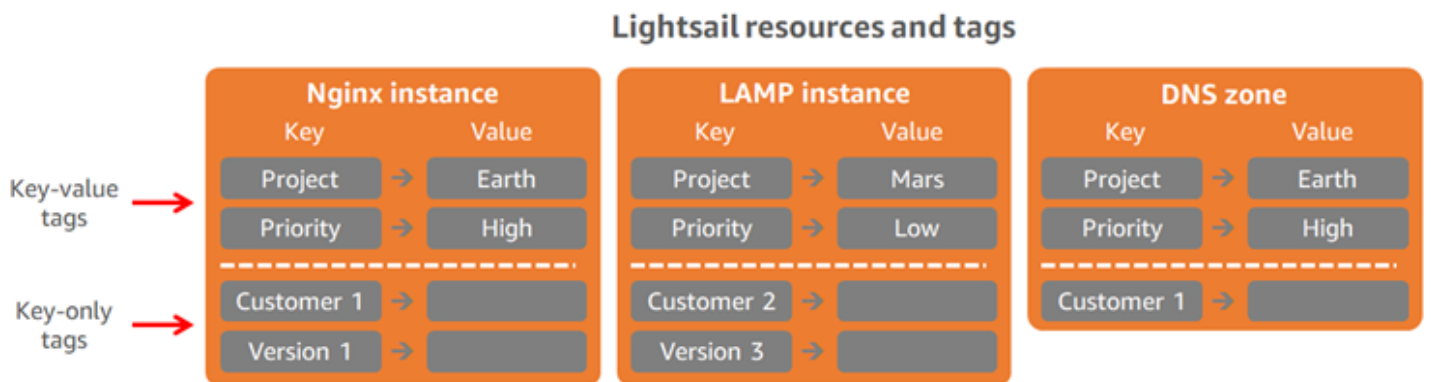
- Deleting notification contacts stops email and SMS text messaging notifications, but it does not stop notification banners from displaying in the Lightsail console. To stop notification banners, and to also stop email and SMS text messaging notifications, disable or delete the alarms that are causing them. For more information, see [Delete or disable metric alarms](#).
- Add your email address and mobile phone number in Lightsail as notification contacts to start receiving email and SMS text messaging notifications again. For more information, see [Add notification contacts](#).

Tags in Amazon Lightsail

With Amazon Lightsail, you can assign labels to your resources as tags. Each tag is a label consisting of a key and an optional value that can make it more efficient to manage, search for, and filter resources.

With Amazon Lightsail, you can assign labels to your resources as tags. Each tag is a label consisting of a key and an optional value that can make it efficient to manage, search for, and filter resources. Although there are no inherent types of tags, they let you categorize Lightsail resources by purpose, owner, environment, or other criteria. This is useful when you have many resources of the same type. You can quickly identify a specific resource based on the tags you've assigned to it. For example, define a set of tags for your resources that helps you track each resource's project, or priority.

A key without a value is referred to as a key-only tag in Lightsail. A key with a value is referred to as a key-value tag. The following diagram illustrates how tagging works. In this example, each resource has a set of key-value and key-only tags. The key-value tags identify projects and priorities, and key-only tags identify customers and application versions.



Use tags to organize billing and control access

You can also use tags to organize your billing, control access to resources and requests in Lightsail, and control access to tag keys. For more information, see one the following guides:

- [Use tags to organize resource costs](#)
- [Use tags to control resource access](#)

Lightsail resources that support tagging

You can tag most Lightsail resources when you create them, or after they are created. If tags cannot be applied during resource creation, Lightsail rolls back the resource creation process. This helps to ensure that resources are either created with tags or not created at all, and that no resources that should be tagged are left untagged at any time.

The following Lightsail resources can be tagged in the Lightsail console:

- Instances
- Container services
- Content delivery network (CDN) distributions
- Buckets
- Databases
- Disks
- DNS zones
- Load balancers

Important

Snapshots created using the Lightsail console automatically inherit tags from the source resource. A Lightsail resource created from that snapshot will have the same tags that were present on the source resource when the snapshot was created.

The following resources can be tagged using the [Lightsail API](#), [AWS Command Line Interface \(AWS CLI\)](#), or SDKs:

- Database snapshots
- Databases
- Disk snapshots
- Disks
- Domains (DNS zones)
- Instance snapshots

- Instances
- Key pairs
- Load balancer TLS certificates (TLS certificates created using Lightsail)
- Load balancers

Important

Snapshots created using the Lightsail API, AWS CLI, or SDKs do not automatically inherit tags from the source resource. Instead, you must manually specify the tags from the source resource using the `tags` parameter.

Tag restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource – 50.
- For each resource, each tag key must be unique. Each tag key can have only one value.
- Maximum key length – 128 Unicode characters in UTF-8.
- Maximum value length – 256 Unicode characters in UTF-8.
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces, and the following characters: `+ - = . _ : / @`
- Tag keys and values are case-sensitive.
- Don't use the `aws :` prefix for either keys or values. That prefix is reserved for AWS use.

Add Lightsail resource tags

Use tags in Amazon Lightsail to categorize your resources by purpose, owner, environment, or other criteria. Tags can be added to resources at or after they are created. Follow these steps to add tags to a resource after it's been created.

Note

For more information about tags, what resources can be tagged, and the restrictions, see [Tags](#).

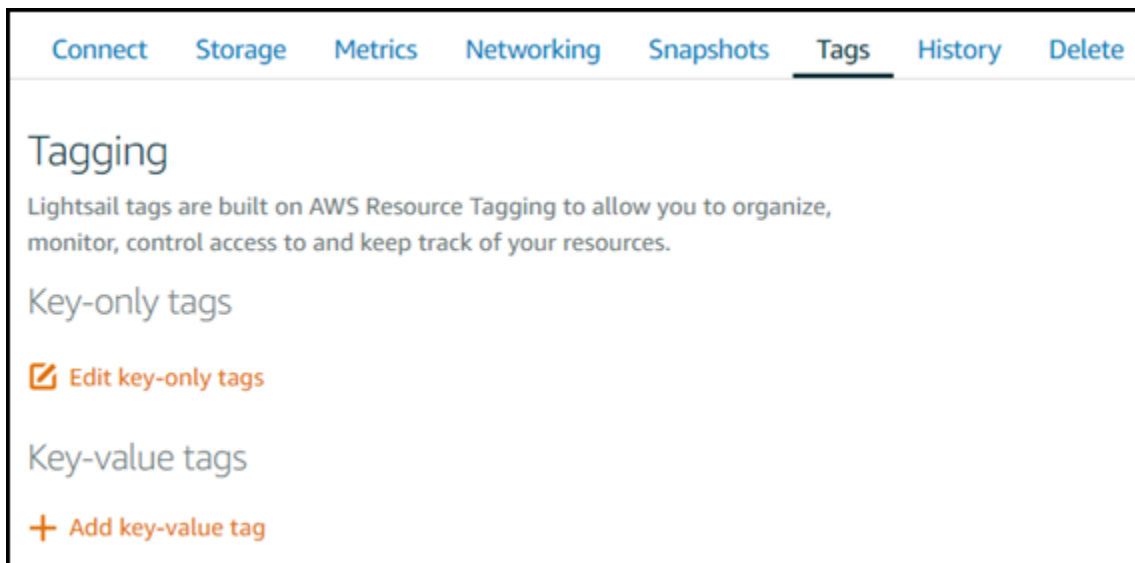
To add tags to a resource

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the tab for the resource type that you want to tag. For example, to add a tag to a DNS zone, choose the **Networking** tab. Or choose the **Instances** tab to add a tag to an instance.

Note

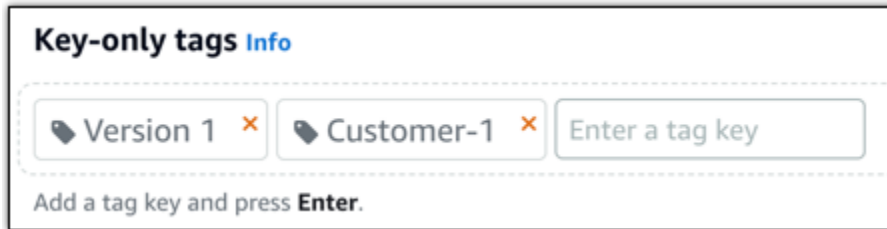
Instances, container services, CDN distributions, buckets, databases, disks, DNS zones, and load balancers can be tagged using the Lightsail console. However, more Lightsail resources can be tagged using the [Lightsail API operations](#), or the [AWS Command Line Interface](#) (AWS CLI) or SDKs. For a full list of Lightsail resources that support tagging, see [Tags](#).

3. Choose the resource that you want to tag.
4. On the management page for the resource that you selected, choose the **Tags** tab.



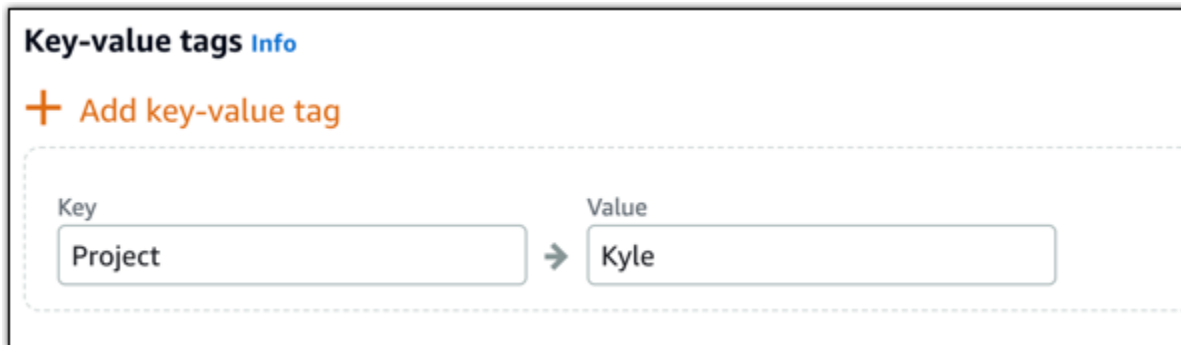
5. Choose one of the following options, depending on the type of tag that you want to add:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Next steps

For more information about tasks that you can perform after adding tags to a resource, see the following guides:

- [Use tags to organize your resources](#)
- [Use tags to organize costs for your resources](#)
- [Use tags to control access to your resources](#)
- [Delete tags](#)

Delete tags in Lightsail

You can delete tags from an Amazon Lightsail resource. Deleting a tag from one resource does not delete the same tag from all other resources. To completely delete a tag from all resources, you must remove that tag from each resource. This guide provides the steps to delete tags from a resource.

Note

For more information about tags, what resources can be tagged, and the tag restrictions, see [Tags](#).

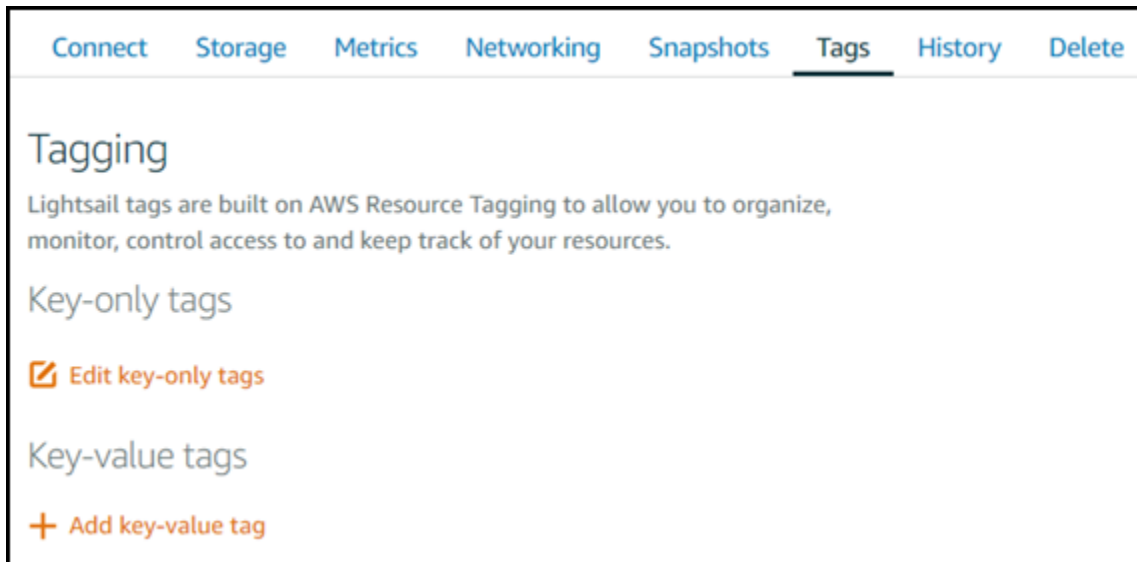
To delete tags from a resource

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the tab for the resource type that you want to delete tags from. For example, to delete tags from a DNS zone, choose the **Networking** tab. Or choose the **Instances** tab to delete tags from an instance.

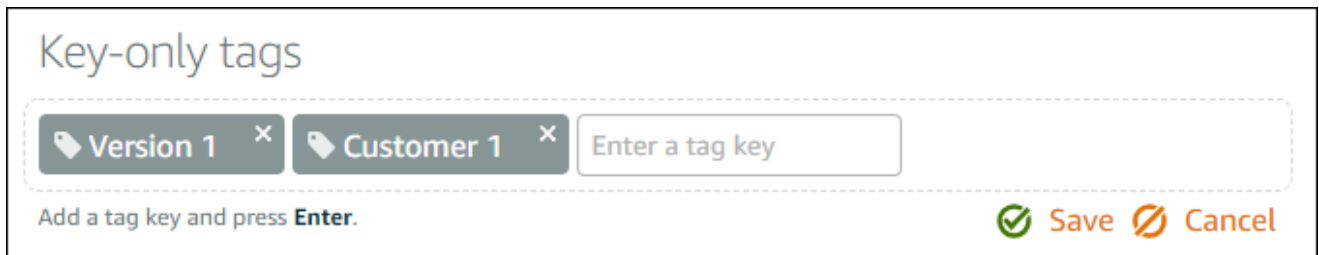
Note

Instances, container services, CDN distributions, buckets, databases, disks, DNS zones, and load balancers can be tagged using the Lightsail console. However, more Lightsail resources can be tagged using the [Lightsail API operations](#), or the [AWS Command Line Interface](#) (AWS CLI) or SDKs. For a full list of Lightsail resources that support tagging, see [Tags](#).

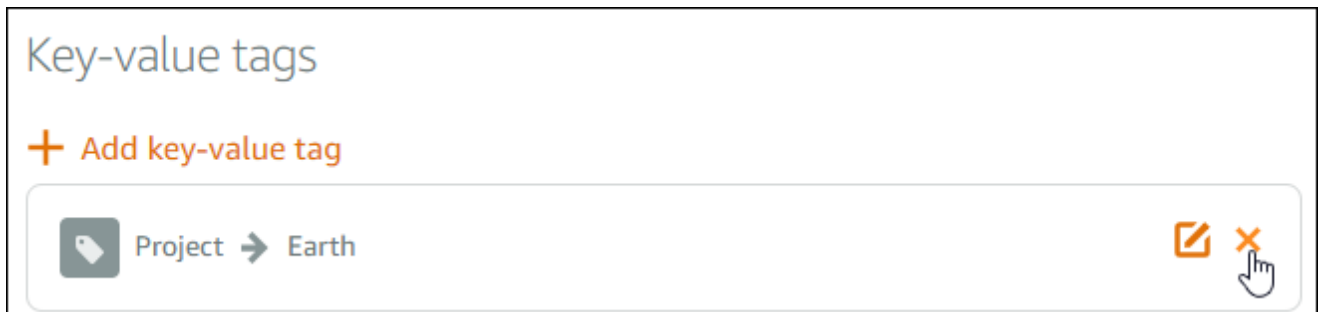
3. Choose the resource that you want to delete tags from.
4. On the management page for the resource you selected, choose the **Tags** tab.



5. Do one of the following, depending on the type of tag that you want to delete from the resource:
 - a. Choose **Edit key-only tags**, then choose the delete icon (X) for the tag that you want to delete from the resource. Choose **Save** when you're done deleting tags to remove them from the resource, or choose **Cancel** to not remove them.



- b. To remove a key-value tag, choose the delete icon (X) for the key-value tag. At the prompt, choose **Yes, delete** to remove the key-value tag, or choose **No, cancel** to not remove it.



Support for resource-level permissions and authorization based on Lightsail tags

Lightsail supports resource-level permissions and authorization based on tags for some of its API actions. For more information, see [Actions, resources, and condition keys for Amazon Lightsail](#) in the *Service Authorization Reference*.

Use tags to control Lightsail resource access

You can use tags in Amazon Lightsail to control access to resources, control access to requests, and control access to tag keys. In this guide, you'll learn how to create an AWS Identity and Access Management (IAM) policy that specifies a key-value tag required to create or delete Lightsail resources, and attach the policy to users or groups who need to make those requests.

Note

To learn more about tags in Lightsail, what resources can be tagged, and the restrictions, see [Tags](#).

Step 1: Create an IAM policy

First, create the following IAM policies in the IAM console. For more information about creating IAM policies, see [Creating IAM Policies](#) in the IAM documentation.

The following policy restricts users from creating new Lightsail resources unless a key tag of `allow` and a value of `true` is defined with the create request. This policy also restricts users from deleting resources unless they have the `allow/true` key-value tag.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lightsail:Create*",
        "lightsail:TagResource",
```



```

        "lightsail:UntagResource"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/allow": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "lightsail:Delete*",
        "lightsail:TagResource",
        "lightsail:UntagResource"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/allow": "true"
        }
    }
}
]
}

```

The following policy restricts users from changing the tag for resources that have a key-value tag that is not allow/false.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "lightsail:TagResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:ResourceTag/allow": "false"
                }
            }
        }
    ]
}

```

```
}  
  }  
    }  
  ]  
}
```

Step 2: Attach the policy to users or groups

After you've created the IAM policies, attach them to the users or groups who need to create Lightsail resources using the key-value pair. For more information about attaching IAM policies to users or groups, see [Adding and Removing IAM Policies](#) in the IAM documentation.

Use tags to organize Lightsail resource costs

You can use tags in Amazon Lightsail to organize your AWS billing to reflect your own cost structure. To do this, add key-value tags to your Lightsail resources. Then activate those tags in the AWS Billing and Cost Management console. Finally, sign up to get your AWS account bill with the tag key values included in your cost allocation report. This guide provides the steps to set this up.

Note

For more information about tags in Lightsail, what resources can be tagged, and tag restrictions, see [Tags](#).

Important

Lightsail database snapshots cannot be tracked in the cost allocation report at this time, even after a cost allocation tag is added to them.

Step 1: Add key-value tags to resources

Add key-value tags to the Lightsail resources that you want to organize in your billing console. For more information about key-value tags, see [Add tags to a resource](#).

It's a good idea to devise a set of tag keys that represent how you want to organize your costs. Your cost allocation report displays the tag keys as additional columns with the applicable values for

each row. Therefore, it's more efficient to track your costs if you use a consistent set of tag keys. For example, you can tag several Lightsail resources with a specific cost center. You do this with a "Cost center" key and a numerical value pairing. Then organize your billing information to see the billing for that cost center across several resources. The following example shows key-value tags that could be used to organize cost allocation:

Key-value tags for cost centers		Key-value tags for projects		Key-value tags for country	
Key	Value	Key	Value	Key	Value
Cost center	→ 5465	Project	→ Earth	Country	→ United States
Cost center	→ 5472	Project	→ Mars	Country	→ England
Cost center	→ 5481	Project	→ Jupiter	Country	→ Paris
Cost center	→ 5486	Project	→ Saturn	Country	→ Japan

Step 2: Activate user-defined cost allocation tags

After you add the necessary tags to your Lightsail resources, activate them for cost allocation in the Billing and Cost Management console. For example, if you created a "Cost center" key tag, then activate that key tag in the Billing and Cost Management console to generate cost-allocation reports for that tag. For more information, see [Activating user-defined cost allocation tags](#) in the AWS Billing and Cost Management documentation.

Step 3: Set up the cost allocation report, and view it

The monthly cost allocation report lists the AWS usage for your account by product category and linked account user. The report contains the same line items as your detailed billing report and additional columns for your tag keys. To set up the monthly cost allocation report, see [Setting up a monthly cost allocation report](#) in the AWS Billing and Cost Management documentation.

When you set up the cost allocation report, you defined an Amazon Simple Storage Service (Amazon S3) bucket where the report is saved. Open the Amazon S3 bucket that you defined and open the cost allocation report after it becomes available. For more information about the contents of the cost allocation report, see [Viewing a cost allocation report](#) in the AWS Billing and Cost Management documentation.

Use tags to organize your Lightsail resources

After you tag your Amazon Lightsail resources, you can filter your resources by the tags you have added. You do this in the Lightsail console by choosing or searching for a tag. This guide shows you how to view and filter your Lightsail resources by tags.

Note

For more information about tags, what resources can be tagged, and tag restrictions, see [Tags](#).

View tags for a resource

Instances, container services, CDN distributions, buckets, databases, disks, DNS zones, and load balancers can be tagged using the Lightsail console and therefore contain a **Tags** tab. That tab is accessible through the resource's management page, as shown in the following example for an instance resource. On the **Tags** tab, you can add, edit, or delete tags. For more information, see [Add tags to a resource](#), and [Delete tags](#).

[Connect](#) [Storage](#) [Metrics](#) [Networking](#) [Snapshots](#) [Tags](#) [History](#) [Delete](#)

Tagging

Lightsail tags are built on AWS Resource Tagging to allow you to organize, monitor, control access to and keep track of your resources.

Key-only tags

🏷️ Version 1 🏷️ Customer 1

[✎ Edit key-only tags](#)

Key-value tags

[+ Add key-value tag](#)

🏷️ Project → Earth ✎ ✕

🏷️ Priority → High ✎ ✕

Note

Instances, container services, CDN distributions, buckets, databases, disks, DNS zones, and load balancers can be tagged using the Lightsail console. However, more Lightsail resources can be tagged using the [Lightsail API operations](#), or the [AWS Command Line Interface \(AWS CLI\)](#) or SDKs. For a full list of Lightsail resources that support tagging, see [Tags](#).

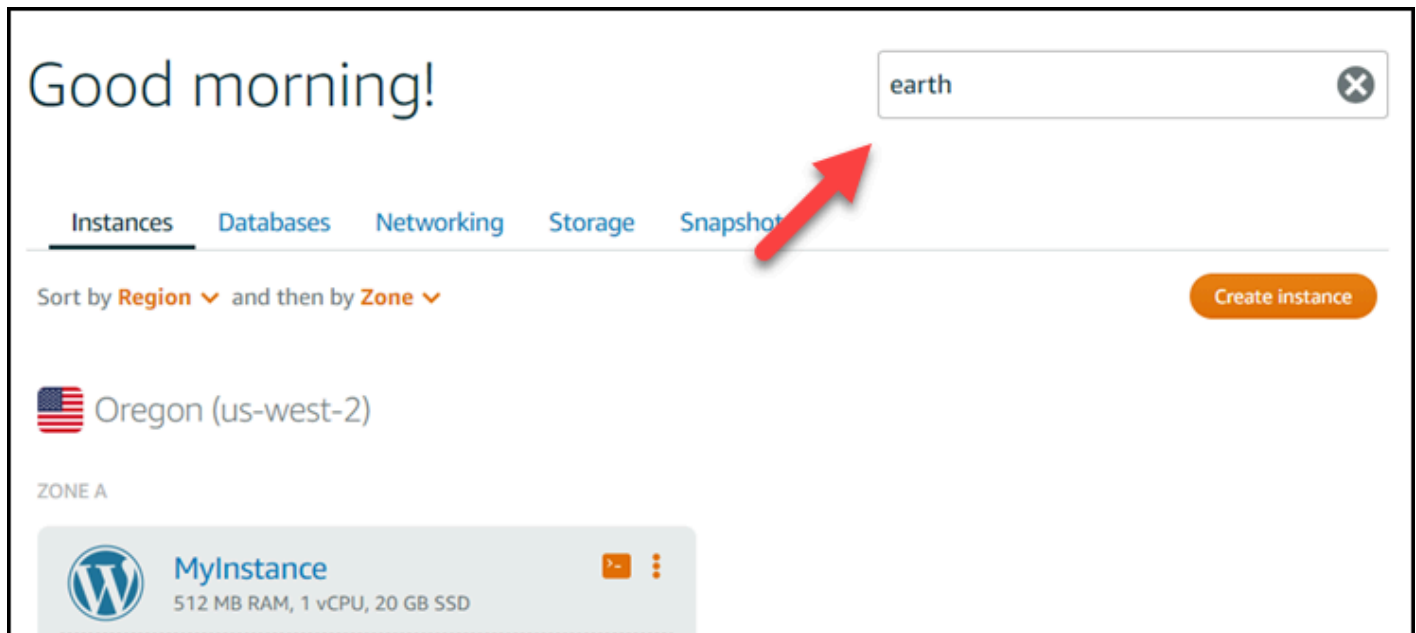
Filter resources using tags

The following options are available in the Lightsail console to filter your resources using tags. All of these options refresh the Lightsail home page to display only the tag that you searched for or selected.

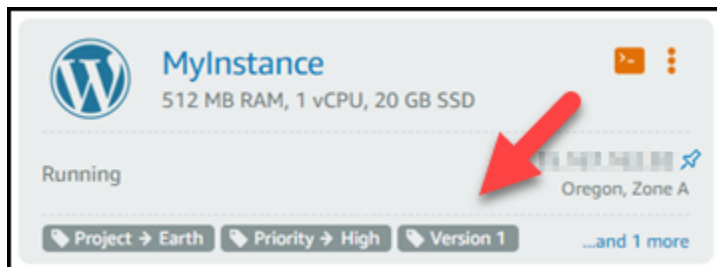
Note

These filtering options are persistent. If you filter by a tag, and then navigate between sections of the Lightsail home page, the filter is still applied.

- On the Lightsail home page, enter the key-only tag or the value that you want to filter by into the **Search** text box, and press **Enter**.



- Choose a tag that is displayed under a resource on the Lightsail home page.



- Choose a tag that is displayed in the heading of a resource.



The screenshot displays the Amazon Lightsail console for an instance named "MyInstance". The instance details include 512 MB RAM, 1 vCPU, and 20 GB SSD, running WordPress in the Oregon, Zone A (us-west-2a) region. A red arrow points to the "Project" tag in the list of tags. The instance status is "Running". The console also shows options to "Stop" or "Reboot" the instance, and a navigation menu with "Connect", "Storage", "Metrics", "Networking", "Snapshots", "Tags", "History", and "Delete".

MyInstance

512 MB RAM, 1 vCPU, 20 GB SSD
WordPress
Oregon, Zone A (us-west-2a)

Project → Earth Priority → High Version 1 Customer 1

Manage 4 tags

Status: **Running**

Private IP: [REDACTED] Static IP: [REDACTED]

Connect Storage Metrics Networking Snapshots Tags History Delete

Troubleshoot Amazon Lightsail resources

The following topics can help you troubleshoot problems that you might encounter with your Amazon Lightsail resources.

Topics

- [Troubleshooting WordPress setup in Lightsail](#)
- [Troubleshoot a 403 \(unauthorized\) error in Lightsail](#)
- [Troubleshoot Lightsail disk issues](#)
- [Troubleshoot connection issues with the Lightsail browser-based SSH or RDP client](#)
- [Troubleshoot a 503 service unavailable error for a Ghost instance in Lightsail](#)
- [Troubleshoot Identity and Access Management \(IAM\) in Lightsail](#)
- [Verify IPv6 reachability in Lightsail](#)
- [Insufficient instance capacity error in Lightsail](#)
- [Troubleshoot Lightsail load balancers](#)
- [Troubleshoot notifications in Lightsail](#)
- [Troubleshoot SSL/TLS certificates in Lightsail](#)

Troubleshooting WordPress setup in Lightsail

Two types of error messages can appear during the WordPress setup workflow in Amazon Lightsail:

Common errors

These types of errors occur immediately after you choose **Create certificate** in the final step of the workflow. These errors will appear in a banner at the top of the Lightsail console. They're typically caused by running the setup workflow on older WordPress instances, or by submitting incorrect information. For example, selecting a DNS record that doesn't point to the public IP address of your instance.

Setup failures

These types of errors occur within a few minutes after you complete the final step in the workflow. These failure messages will appear in the **Set up your WordPress website** section of the instance **Connect** tab. These errors happen when the Let's Encrypt HTTPS certificate cannot be configured on your instance.

Use the information in the following topics to help you diagnose and fix any errors that you might encounter with the WordPress setup guided workflow.

Topics

- [Troubleshooting WordPress setup common errors in Lightsail](#)
- [Troubleshooting WordPress setup failures in Lightsail](#)

For more information about the WordPress setup guided workflow in Amazon Lightsail, see [Configure your WordPress instance](#).

Troubleshooting WordPress setup common errors in Lightsail

An error message will appear at the top of the Lightsail console if there's an issue with the information that was submitted during the workflow.

The first line of the message informs you that setup has encountered an error:

Could not complete setup on your instance *InstanceName* in the *InstanceRegion* Region.

The second line contains the error that setup encountered:

An error occurred and we were unable to connect or stay connected to your instance

We encountered an error while configuring the Let's Encrypt SSL/TLS certificate on your instance test-2 in the us-east-1 Region. Try again later.
An error occurred and we were unable to connect or stay connected to your instance. If this instance has just started up, try again in a minute or two.

To begin troubleshooting, match the error that appeared in the message with one of the following errors.

Errors

- [DNS records not found. Confirm that the domain's DNS records point to the public IP address of your instance, and allow time for DNS changes to propagate.](#)
- [DNS records do not match. Confirm that the domain's DNS records point to the public IP address of your instance, and allow time for DNS changes to propagate.](#)
- [Unable to connect to your instance. Allow a few minutes for the SSH connection to become ready. Then, start setup again.](#)
- [Unsupported WordPress version. Setup only supports WordPress versions 6, and up.](#)
- [Setup only supports WordPress instances that were created on or after January 1, 2023.](#)

- [Instance firewall ports 22, 80, and 443 must allow a TCP connection from any IP address during the setup workflow. You can change these settings from the instance Networking tab.](#)

DNS records not found. Confirm that the domain's DNS records point to the public IP address of your instance, and allow time for DNS changes to propagate.

Reason

This error is caused by misconfigured DNS records, or DNS records that have not had sufficient time to propagate throughout the Internet's DNS.

Fix

Confirm that the **A** or **AAAA** DNS records are present in the DNS zone, and that they point to the public IP address of your instance. For more information, see [DNS in Lightsail](#).

When you add or update DNS records that point traffic from your apex domain (example.com) and its www subdomains (www.example.com), they will need to propagate throughout the Internet's DNS. You can verify that your DNS changes have taken effect by using tools such as [nslookup](#), or [DNS Lookup](#) from *MxToolbox*.

Note

Allow time for any DNS record changes to propagate through the internet's DNS, which may take several hours.

DNS records do not match. Confirm that the domain's DNS records point to the public IP address of your instance, and allow time for DNS changes to propagate.

Reason

The **A** or **AAAA** DNS records do not point to the public IP address of the instance.

Fix

Confirm that the **A** or **AAAA** DNS records are present in the DNS zone, and that they point to the public IP address of your instance. For more information, see [DNS in Lightsail](#).

Note

Allow time for any DNS record changes to propagate through the internet's DNS, which may take several hours.

Unable to connect to your instance. Allow a few minutes for the SSH connection to become ready. Then, start setup again.**Reason**

The instance was just created or rebooted, and the SSH connection is not ready.

Fix

Allow a few minutes for the SSH connection to become ready. Then, retry the guided workflow. For more information, see [Troubleshooting SSH in Lightsail](#).

Unsupported WordPress version. Setup only supports WordPress versions 6, and up.**Reason**

The version of WordPress that's installed on the instance is older than WordPress version 6. Older WordPress versions contain incompatible software and dependencies that prevent the HTTPS certificate from being generated.

Fix

Create a new WordPress instance from the Lightsail console. Then, migrate the WordPress website from the older instance to the new one. For more information, see [Migrate an existing WordPress blog](#).

If you're creating a new instance to replace the existing instance, make sure to update your application dependencies to your new instance.

Setup only supports WordPress instances that were created on or after January 1, 2023.

Reason

The instance that is being used with setup, might contain outdated software. Older software will prevent the HTTPS certificate from being generated.

Fix

Create a new WordPress instance from the Lightsail console. Then, migrate the WordPress website from the older instance to the new one. For more information, see [Migrate an existing WordPress blog](#).

If you're creating a new instance to replace the existing instance, make sure to update your application dependencies to your new instance.

Instance firewall ports 22, 80, and 443 must allow a TCP connection from any IP address during the setup workflow. You can change these settings from the instance Networking tab.

Reason

Instance firewall ports 22, 80 and 443 must allow TCP connections from any IP address while setup is running. This error is generated when one or more of these ports are closed. For more information, see [Instance firewalls](#).

Fix

Add or edit the instance's IPv4 and IPv6 firewall rules to allow TCP connections over ports 22, 80, and 443. For more information, see [Add and edit instance firewall rules](#).


Troubleshooting WordPress setup failures in Lightsail

Setup failure messages appear in the **Set up your WordPress website** section of the instance **Connect** tab. Setup failures can occur within a few minutes after you complete the final step in the workflow. They're caused when the Let's Encrypt HTTPS certificate cannot be configured on your instance.

Failed to complete setup – Review the following status messages, and restart setup to update your configuration. Download the error log for more details.

⊗ Failed to complete setup
Review the following status messages, and restart setup to update your configuration.
[Download the error log](#) for more details.

[Restart setup](#)



- ✔ Domain
- ✔ DNS zone
- ✔ Static IP
- ✔ Map domains & subdomains
- ⊗ **SSL/TLS certificate**
Certificate failed to validate.

From the failure message, choose the **Download the error log** link to download and view the error logs that setup generated. To begin troubleshooting, match the error message from the logs with one of the following errors.

Errors

- [Certbot.errors.AuthorizationError: Some challenges have failed](#)
- [Certbot failed to authenticate some domains](#)
- [Too many certificates \(5\) already issued for this exact set of domains in the last 168 hours](#)
- [Too many failed authorizations](#)

Certbot.errors.AuthorizationError: Some challenges have failed

Reason

This error is caused by misconfigured DNS records, or DNS records that have not had sufficient time to propagate throughout the Internet.

Fix

Verify that the **A** or **AAAA** DNS records are present in the DNS zone, and that they point to the public IP address of your instance. For more information, see [DNS in Lightsail](#).

When you add or update DNS records that point traffic from your apex domain (example.com) and its www subdomains (www.example.com), they will need to propagate throughout the Internet. You can verify that your DNS changes have taken effect by using tools such as [nslookup](#), or [DNS Lookup](#) from *MxToolbox*.

Note

Allow time for any DNS record changes to propagate through the internet's DNS, which may take several hours.

Certbot failed to authenticate some domains

Reason

This error can surface if another process is using port 80 while the HTTPS certificate is being configured on the instance.

Fix

Restart your WordPress instance. Then, run the guided workflow again. Use the following procedure to terminate any running processes on the instance that are running on port 80 if restarting doesn't resolve the issue.

Procedure

1. Connect to your instance by using the Lightsail [browser-based SSH client](#), or by using [AWS CloudShell](#).
2. Stop the Bitnami process that's running on the instance:

```
$ sudo /opt/bitnami/ctlscript.sh stop
```

Verify that the Bitnami process is stopped:

```
sudo /opt/bitnami/ctlscript.sh status
```

3. Check if there are other processes that are using port 80:

```
fuser -n tcp 80
```

4. Terminate any processes that are not needed by another application:

```
fuser -k -n tcp 80
```

5. Restart WordPress setup.

Too many certificates (5) already issued for this exact set of domains in the last 168 hours

Reason

One or more of your domains or subdomains has already been used to create 5 certificates within the last week. For more information, see [Rate Limits](#) on the *Let's Encrypt website*.

Fix

Wait one week (168 hours), and then restart the guided workflow for this domain.

Too many failed authorizations

Reason

One or more of the domains or subdomains in the request has exceeded the limit of five validations per hour. For more information, see [Rate Limits](#) on the *Let's Encrypt website*.

Fix

Wait one hour and run WordPress setup again. Verify that other validation errors have been fixed before you restart setup.

Troubleshoot a 403 (unauthorized) error in Lightsail

If you get a 403 error when trying to access the [Lightsail console](#), don't panic. Try these steps to troubleshoot the problem:

- If your AWS account or your AWS Identity and Access Management (IAM) user was recently created, wait a few minutes, and then refresh your browser.
- If it's been a while since you last signed in, refresh your browser. If you're prompted to sign in again, be sure to use an IAM user that has access to Lightsail.
- If your IAM user doesn't have access to Lightsail, then contact the [AWS account root user](#) or an IAM user with administrator access to request access to Lightsail. To learn more, see [Manage access to Amazon Lightsail for an IAM user](#).
- If you continue to get the 403 error after trying the above steps, contact [AWS Support](#). In some rare cases for AWS accounts created before 2011, support will have to manually subscribe your account to Lightsail.

Troubleshoot Lightsail disk issues

You might encounter errors with your block storage disks in Lightsail. This topic identifies common issues and workarounds for those errors.

General disk errors

Choose the issue below that best describes your problem, and follow the links to fix the issue. If you encounter an issue that's not in the list, use the [Questions? Comments?](#) link at the bottom of this page to submit feedback or contact [AWS Support](#).

I can't delete a disk because it's still attached to an instance.

Try detaching the disk from your instance first, and then try to delete the disk. For more information, see [Detach and delete a block storage disk](#).

Actual error message: **You can't perform this operation because the disk is still attached to a Lightsail instance: *YOUR_INSTANCE***

My disk has a status of error.

The **error** status indicates that the underlying hardware related to your Lightsail disk has failed. You can restore the disk from a recent snapshot, otherwise the data associated with the disk is unrecoverable. For more information, see [Create a block storage disk from a snapshot](#).

You are not billed for disks with a status of **error**.

I can't detach a disk because the Lightsail instance is still running.

Try stopping your instance first, and then try to detach the disk. For more information, see [Stop an instance](#).

Actual error message: **You can't detach this disk right now. The state of this disk is: **DISK_STATE****

I can't specify a custom disk size above 16 TB (16,384 GB).

Try creating a smaller disk. Additional disks can be up to 16 TB. If your disk is less than 16 TB and you still can't create it, you might encounter the next error in the list (too many big disks). That's because you can't have more than 20 TB in additional disk storage across your AWS account. For more information, see [Block storage disks](#).

Actual error message: **The size of a block storage disk must be between 8 and 16384 GB.**

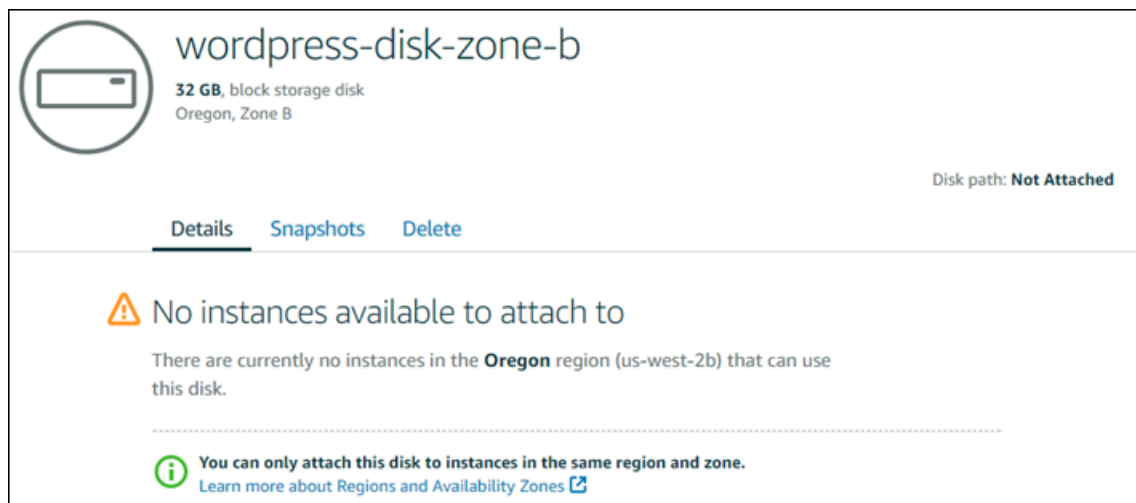
I can't create any more disks in Lightsail.

You might have reached your quota for the number of disks you can create. Or you might have created too many big disks (the total size of disk storage can't exceed 20 TB) in your AWS account. For more information, see [Block storage disks](#).

Actual error message: **You've reached the maximum size limit of all disks in this account. or You've reached the limit of disks in this account.**

I can't attach my disk to my Lightsail instance

If you encounter the following error, you need to recreate your disk in the same AWS Region and Availability Zone as the instance where you plan to attach the disk.



wordpress-disk-zone-b
32 GB, block storage disk
Oregon, Zone B

Disk path: **Not Attached**

[Details](#) [Snapshots](#) [Delete](#)

⚠️ No instances available to attach to
There are currently no instances in the **Oregon** region (us-west-2b) that can use this disk.

i You can only attach this disk to instances in the same region and zone.
[Learn more about Regions and Availability Zones](#)

Actual error message: **There are currently no instances in the *AWS Region* that can use this disk.**

Troubleshoot connection issues with the Lightsail browser-based SSH or RDP client

You might get an error message when trying to connect to an instance using the browser-based SSH or RDP clients available in the Amazon Lightsail console. The possible reasons for this error are discussed in the following sections.

Important

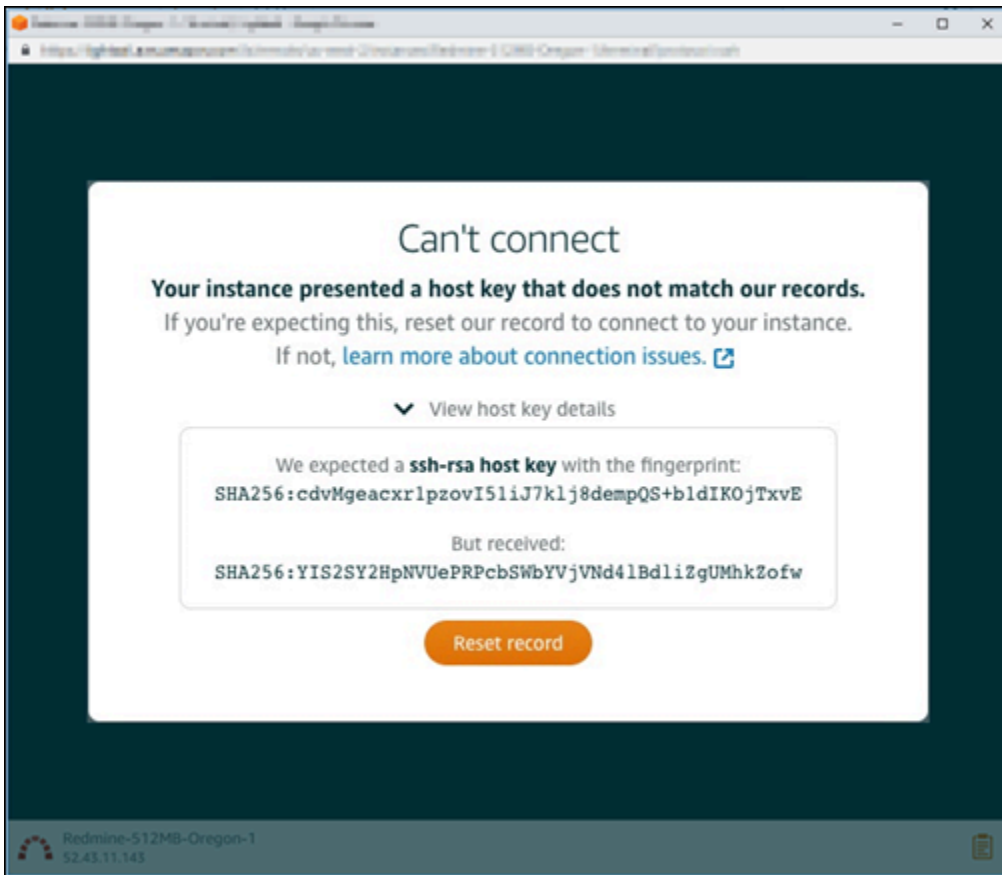
The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

Error message: Can't connect

The SSH and RDP browser-based clients use host key or certificate validation to authenticate an instance when trying to connect to it. If the instance presents a host key or certificate that doesn't match the one that Lightsail has on record, one of two error messages display. Both error messages are shown and described in this section.

Can't connect, reset record

The following error message displays when there's a host key or certificate mismatch, and Lightsail determines that the mismatch might have been caused by a recent operating system upgrade, or a deliberate update to the host key or certificate by you or another user. In this case, Lightsail has determined that the host key or certificate mismatch was not caused by a bad actor on the network between your browser and the instance.



Choose **Reset record** if you expected the mismatch. This action deletes the host key or certificate that Lightsail has on record for the instance, and permits the browser-based SSH or RDP session to connect to the instance.

You can also delete the host key or certificate that Lightsail has on record by using the following AWS Command Line Interface (AWS CLI) command. For *InstanceName*, enter the name of your instance for which you want to delete the known host key or certificate. For *Region*, enter the AWS Region of the instance.

```
aws lightsail delete-known-host-keys --region Region --instance-name InstanceName
```

Example:

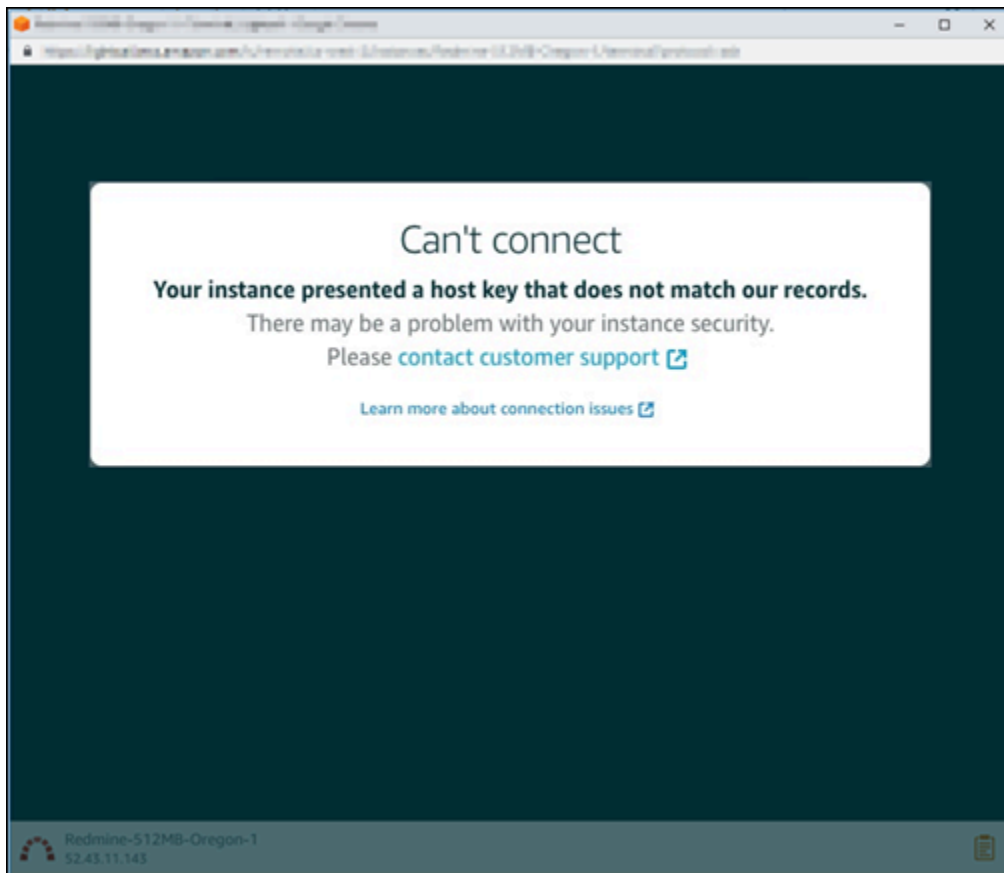
```
aws lightsail delete-known-host-keys --region us-west-2 --instance-name WordPress-512MB-Oregon-1
```

Note

For more information about the AWS CLI, see [Configure the AWS CLI to work with Lightsail](#).

Can't connect, contact customer support

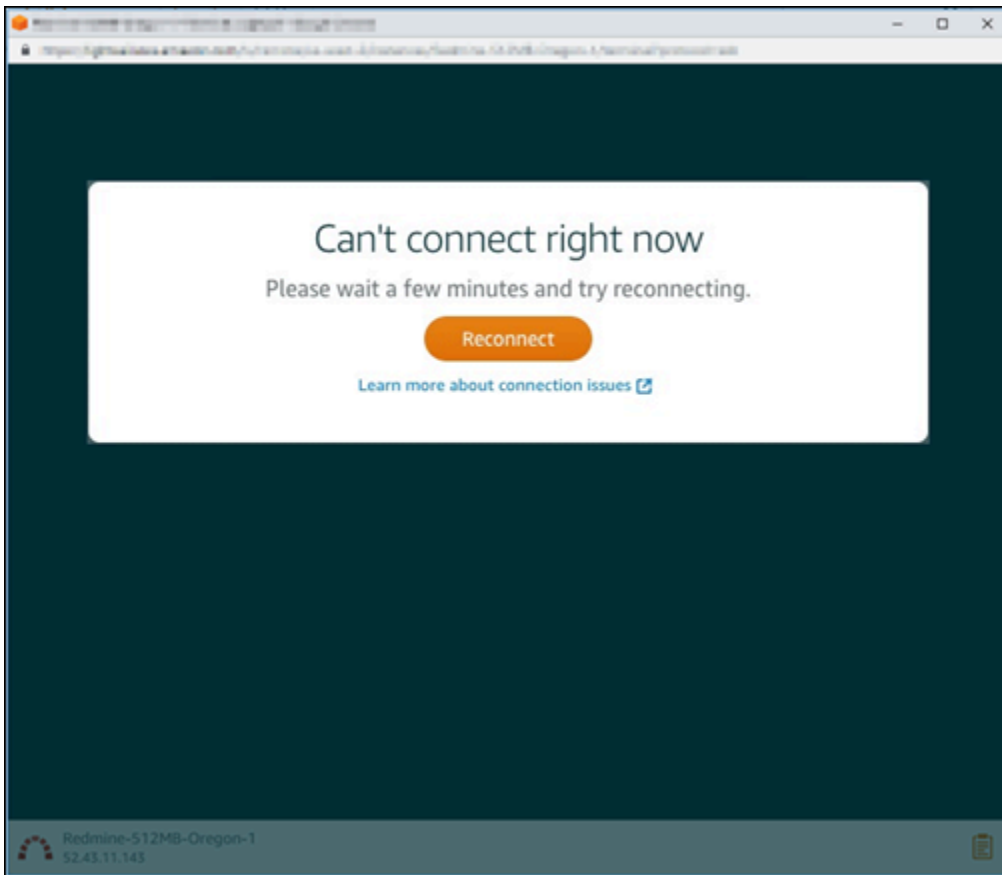
The following error message displays when there's a host key or certificate mismatch, and Lightsail determines that there is suspicious activity that warrants further investigation, such as a man-in-the-middle attack.



This error message means that you can't connect to the instance using the browser-based SSH or RDP client. [Contact support](#) for assistance.

Error message: Can't connect right now

The following error message displays when you try to connect to an instance that hasn't yet started after it's created, rebooted, or restarted. Wait a few minutes and then choose **Reconnect** to try again.



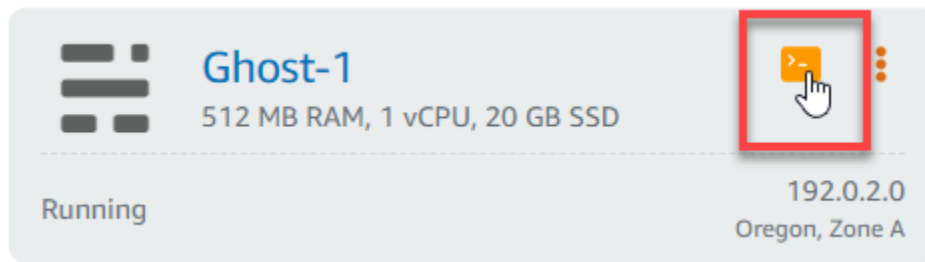
If you still can't connect, [contact AWS Support](#).

Troubleshoot a 503 service unavailable error for a Ghost instance in Lightsail

After you create a new Ghost instance in Amazon Lightsail, and try to access your website, you might see an error stating that the service is unavailable (503). In some cases, the Ghost service on the instance is not automatically started when the instance is created. This can happen when you select the \$3.50 USD/month bundle for your instance. Use the following procedure to start the Ghost service, and resolve the "service is unavailable" error.

Start the Ghost service

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.
3. Choose the browser-based SSH client icon for your Ghost instance.



4. After the SSH client is connected, enter the following command to restart all services on the instance:

```
sudo /opt/bitnami/ctlscript.sh restart
```

You should see a result similar to the following example:

```
bitnami@ip-172-26-11-214:~$ sudo /opt/bitnami/ctlscript.sh restart
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
/opt/bitnami/apps/ghost/scripts/ctl.sh : ghost not running
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql started at port 3306
Ensuring user is not logged in as ghost user [skipped]
Checking if logged in user is directory owner [skipped]
✓ Checking current folder permissions
✓ Validating config
✓ Checking memory availability
✓ Checking binary dependencies
✓ Starting Ghost: 127-0-0-1

-----

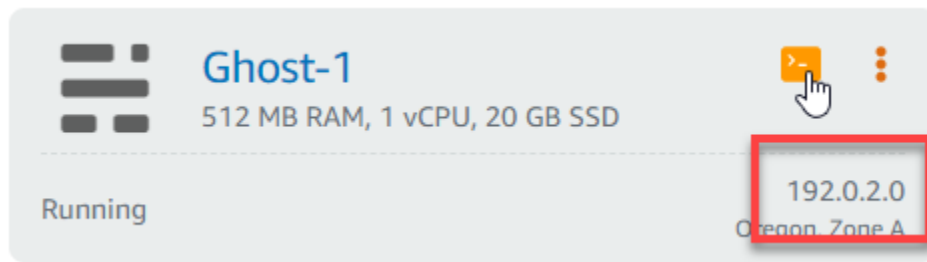
Your admin interface is located at:

  http://18.237.117.48:80/ghost/

/opt/bitnami/apps/ghost/scripts/ctl.sh : ghost started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80
```

5. Browse to the public IP address of your instance to confirm that your Ghost website is up and running.

The public IP address of your instance is listed next to the instance name in the **Instances** tab of the Lightsail console.



When you browse to the public IP of your new Ghost instance, you should see the default Ghost website template:



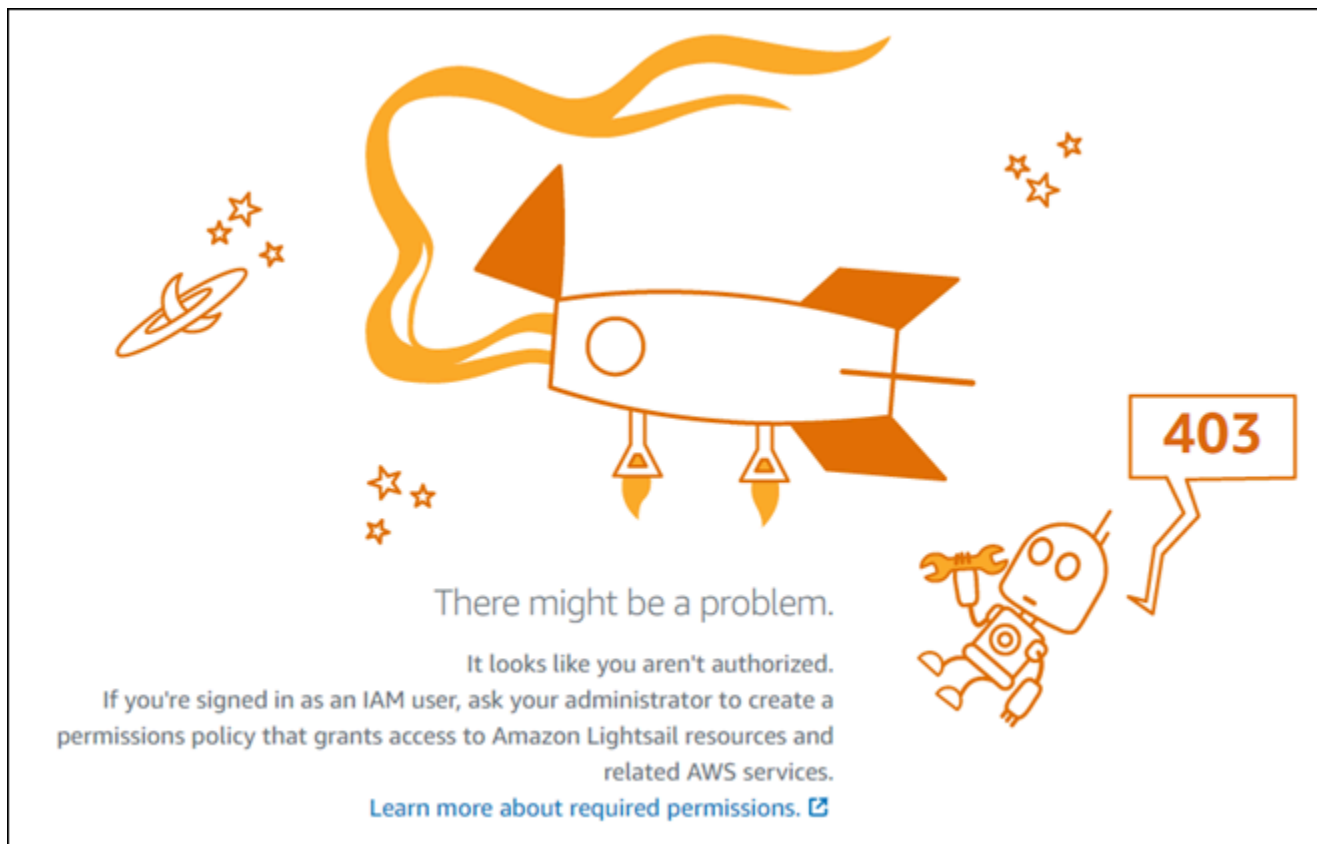
Troubleshoot Identity and Access Management (IAM) in Lightsail

Use the following information to help you diagnose and fix common issues that you might encounter when working with Lightsail and IAM.

I am not authorized to perform an action in Lightsail

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the `mateojackson` IAM user tries to access the Lightsail console but does not have `lightsail:*` (full-access) permissions.



In this case, Mateo asks his administrator to update his policies to allow him to access the Lightsail console using the `lightsail:*` (full-access) permissions.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Amazon Lightsail.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Amazon Lightsail. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

Important

Do not provide your access keys to a third party, even to help [find your canonical user ID](#). By doing this, you might give someone permanent access to your AWS account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see [Managing access keys](#) in the *IAM User Guide*.

I'm an administrator and want to allow others to access Lightsail

To allow others to access Amazon Lightsail, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Amazon Lightsail.

To get started right away, see [Creating your first IAM delegated user and group](#) in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my Lightsail resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Amazon Lightsail supports these features, see [How Amazon Lightsail works with IAM](#).
- To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](#) in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing access to AWS accounts owned by third parties](#) in the *IAM User Guide*.
- To learn how to provide access through identity federation, see [Providing access to externally authenticated users \(identity federation\)](#) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.

Verify IPv6 reachability in Lightsail

You can verify IPv6 connectivity from your local computer to an Amazon Lightsail instance using the ping tool. Ping is a network diagnostic utility that's used to troubleshoot connectivity issues between two or more networked devices. If ping succeeds, you should be able to connect to your instance over IPv6. If a network setting or device isn't configured to allow IPv6, the ping command fails. For more information, see [IPv6 considerations](#)

Contents

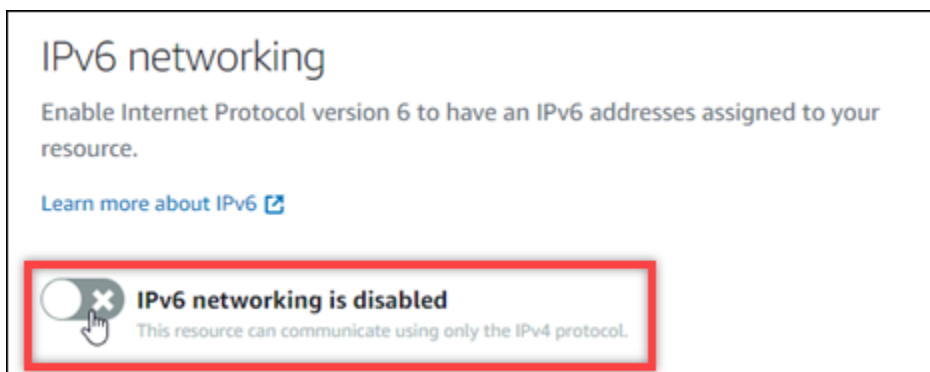
- [Enable IPv6 for dual-stack instances](#)
- [Configure the instance's firewall](#)
- [Test reachability to your instance](#)

Enable IPv6 for dual-stack instances

Enable IPv6 for your dual-stack instance before you begin testing. IPv6 is always on for IPv6-only instances.

Complete the following procedure to enable IPv6 on your dual-stack instance if it's not enabled.

1. Sign in to the [Lightsail console](#).
2. Choose the name of the instance for which you want to enable IPv6. Make sure that your instance is running.
3. Choose the **Networking** tab from the instance management page.
4. Enable IPv6 on the **IPv6 Networking** section of the page.



After you enable IPv6, a public IPv6 address is assigned to your instance, and the IPv6 firewall becomes available.

IPv6 networking is enabled
This resource can communicate using the IPv4 and IPv6 protocols.

PUBLIC IPV6

2001:0db8:85a3:0000:0000:8a2e:0370:7334

The public IPv6 address of your instance changes only when you disable and re-enable IPv6.

IPv6 firewall ?

Create rules to open ports to the internet, or to a specific IPv6 address or range.
[Learn more about firewall rules](#)

+ Add rule

Application	Protocol	Port or range / Code	Restricted to		
SSH	TCP	22	Any IPv6 address		
HTTP	TCP	80	Any IPv6 address		
HTTPS	TCP	443	Any IPv6 address		

- Take note of the instance's **Public IPv4** and **Public IPv6** addresses at the top of the page. You'll use them in the following sections.

Configure the instance's firewall

The firewall in the Lightsail console acts as a virtual firewall. Meaning it controls which traffic is allowed to connect to your instance through its public IP address. Each dual-stack instance that you create in Lightsail has an individual firewall for IPv4 addresses and another for IPv6 addresses. Each firewall contains a set of rules that filter traffic coming into the instance. Both firewalls are independent of each other—you must configure firewall rules separately for IPv4 and IPv6. Instances with an IPv6-only instance plan don't have an IPv4 firewall that you can configure.

Complete the following procedure to configure your instance's firewall for Internet Control Message Protocol (ICMP) traffic. The ping utility uses the ICMP protocol to communicate with your instance. For more information, see [Instance firewalls in Amazon Lightsail](#).

Important

Windows and Linux contain an operating system (OS) level firewall that can block ping commands. Verify that the instance's OS firewall can accept ICMP traffic over IPv4 and IPv6 before you continue. For more information, see the following documentation:

- [Connect to your Lightsail Windows instance](#)
- [Connect to your Lightsail Linux or Unix instances](#)

1. Sign in to the [Lightsail console](#).
2. Choose the name of the instance for which you want to configure the firewall.
3. Choose the **Networking** tab from the instance management page, then complete the remaining steps in the appropriate section for the type of firewall that you want to use. For IPv4, complete the steps in the **IPv4 Firewall** section. For IPv6, complete the steps in the **IPv6 Firewall** section.
 - a. From the **Application** dropdown menu, choose **Ping (ICMP)**.
 - b. Select the **Restrict to IP address** box to allow a connection from your local source IP address or range, then enter your source IP address. (Optional) You can leave the box unselected to allow a connection from any IP address. We recommend that you use this option in a test environment only.
 - c. Choose **Create** to apply the new rule to your instance.

Test reachability to your instance

Complete the following procedure to test IPv4 or IPv6 reachability from your local computer or network to your Lightsail instance. You need the instance's public IPv4 and IPv6 addresses that you noted in [Step 5](#).

From a Linux, Unix, or macOS device

1. Open a terminal window on your local device.
2. Enter one of the following commands to ping your Lightsail instance. Replace the example *IP address* that's in the command with the public IPv4 or IPv6 address of your instance.

To test over IPv4

```
ping 192.0.2.0
```

To test over IPv6

```
ping6 2001:db8::
```

3. After the command returns a few replies, enter `ctrl+z` on your device's keyboard to stop the command.

The `ping` command returns successful replies from your instance's IPv4 address if it's successful. The result should look like the following example.

```
$ ping 192.0.2.0
PING 192.0.2.0 56(84) bytes of data:
64 bytes from 192.0.2.0: icmp_seq=1 ttl=63 time=0.323 ms
64 bytes from 192.0.2.0: icmp_seq=2 ttl=63 time=0.284 ms
64 bytes from 192.0.2.0: icmp_seq=3 ttl=63 time=0.324 ms
64 bytes from 192.0.2.0: icmp_seq=4 ttl=63 time=0.617 ms
^Z
[1]+  Stopped                  ping 192.0.2.0
$
```

The `ping6` command returns successful replies from your instance's IPv6 address if it's successful. The result should look like the following example.

```
$ ping6 2001:db8::
PING 2001:db8:: 56 data bytes
64 bytes from 2001:db8::: icmp_seq=1 ttl=255 time=0.698 ms
64 bytes from 2001:db8::: icmp_seq=2 ttl=255 time=0.228 ms
64 bytes from 2001:db8::: icmp_seq=3 ttl=255 time=0.322 ms
^Z
[1]+  Stopped                  ping6 2001:db8::
```

Both commands return **Request timeout** if your instance can't be reached.

From a Windows device

1. Open a command prompt.
2. Enter one of the following commands to ping your Lightsail instance. Replace the example *IP address* that's in the command with the public IPv4 or IPv6 address of your instance.

To test over IPv4

```
ping 192.0.2.0
```

To test over IPv6

```
ping 2001:db8::
```

3. After the command returns a few replies, enter `ctrl+z` on your device's keyboard to stop the command.

The ping command returns successful replies from your instance's IPv4 address if it's successful. The result should look like the following example.

```
C:\Users\Administrator>ping 192.0.2.0

Pinging 192.0.2.0 with 32 bytes of data:
Reply from 192.0.2.0: bytes=32 time=10ms TTL=53
Reply from 192.0.2.0: bytes=32 time=10ms TTL=53
Reply from 192.0.2.0: bytes=32 time=11ms TTL=53
Reply from 192.0.2.0: bytes=32 time=10ms TTL=53

Ping statistics for 192.0.2.0:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 10ms, Maximum = 11ms, Average = 10ms
```

The ping command returns successful replies from your instance's IPv6 address if it's successful. The result should look like the following example.

```
C:\Users\Administrator>ping 2001:db8::

Pinging 2001:db8:: with 32 bytes of data:
Reply from 2001:db8::: time=74ms
Reply from 2001:db8::: time=74ms
Reply from 2001:db8::: time=74ms
Reply from 2001:db8::: time=74ms

Ping statistics for 2001:db8:::
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 74ms, Maximum = 74ms, Average = 74ms
```

Both commands return **Request timeout** if your instance can't be reached.

Insufficient instance capacity error in Lightsail

You might get an insufficient error when you try to launch an instance or restart a stopped instance. This means that AWS doesn't have the available instance capacity to fulfill your request at the current time. Following is an example of the insufficient instance capacity error:

InsufficientInstanceCapacity: There is not enough capacity to fulfill your instance request. Reduce the number of instances in your request, or wait for additional capacity to become available. You can also try launching an instance by selecting a smaller Lightsail plan (which you can resize at a later stage)."

In this guide, you will learn about the actions you can take if you get an insufficient instance capacity error.

Contents

- [Insufficient capacity when launching a new instance](#)
- [Insufficient capacity when starting a stopped instance](#)
- [Related information](#)

Insufficient capacity when launching a new instance

Use the following options if you get an insufficient instance capacity error when launching a new instance. You can complete each option in order, or choose an option that works for you.

1. Wait a few minutes and then submit your request again. Instance capacity can shift frequently. Continue to option 2 if you are unable to create your instance after waiting a few minutes.
2. Select a different Availability Zone (AZ) when creating your instance. Each AWS Region contains three or more AZs, and each AZ maintains different instance capacities. By selecting a different AZ, you can take advantage of its current instance capacity. Continue to option 3 if you are unable to create an instance in a different AWS Region or AZ.
3. Reduce the number of instances in your request. If you're creating multiple instances at the same time, reduce the number of instances and submit your request again. Continue to option 4 if reducing the number of instances doesn't resolve the issue.
4. Choose a different instance plan when creating your instance. Choose a different instance plan if you are unable to create an instance in a different AZ or Region. You can resize the instance at a later stage. For more information about resizing your instance, see [Create an instance from a snapshot](#).

Insufficient capacity when starting a stopped instance

Use the following options if you get an insufficient instance capacity error when starting an existing instance that was previously stopped.

1. Wait a few minutes and then submit your request again. Instance capacity can shift frequently. Continue to option 2 if you are unable to create your instance after waiting a few minutes.
2. Create a new instance from a snapshot. Take a snapshot of the stopped instance. Then, use the snapshot to create a new instance in an AZ that's different from the original instance. For example, if your instance is currently in us-east-2a (Zone A), select us-east-2c (Zone C) when you create the new instance. For more information, see [Create an instance from a snapshot](#).
3. You can also choose a different instance plan when creating a new instance from a snapshot. This action is optional.

Important

After the new instance is running, verify you have access to the new instance and everything is working properly. For example, if your instance was running an application, make sure that the application is working as expected. If so, you can delete the earlier instance.

Related information

[Frequently asked questions](#)

[Resilience in Lightsail](#)

Troubleshoot Lightsail load balancers

You might encounter errors with your Lightsail load balancers. This topic identifies common issues and workarounds for those errors.

General load balancer errors

Choose the issue below that best describes your problem, and follow the links to fix the issue. If you encounter an issue that's not in the list, use the **Questions? Comments?** link at the bottom of this page to submit feedback or contact AWS Customer Support.

I can't create a certificate.

There is a quota to the number of certificates you can create in an AWS account. For more information, see [Quotas](#) in the AWS Certificate Manager User Guide. The same quota apply to Lightsail certificates for load balancers.

Actual error message: **Sorry, you've requested too many certificates for your account.**

I can't attach any more instances to my load balancer.

You can attach as many Lightsail instances as you like to your load balancer, as long as you stay within the quota of 20 total Lightsail instances per AWS account.

Actual error message: **Sorry, you've reached the maximum number of instances you can attach to this load balancer.**

I can't attach a specific instance to my load balancer.

First, check to make sure your Lightsail instance is running. If it is stopped, you can start it from the instance management page. Lightsail instances must be running to be successfully attached to a load balancer.

It could be that you have attached the same instance to too many load balancers.

Actual error message: **Sorry, you've reached the maximum number of times an instance can be registered with a load balancer.**

Lightsail can't find the instance I'm trying to attach to my load balancer

You might be trying to attach an instance that no longer exists or is not in the same VPC as the target group.

Actual error message: **Sorry, the instance you specified doesn't exist, isn't in the same VPC as the target group, or has an unsupported instance type.**

Troubleshoot notifications in Lightsail

If don't receive notifications when you expect to be notified, then there are a few things you should check to confirm that your notification contacts are configured correctly. To learn more about notifications, see [Notifications](#).

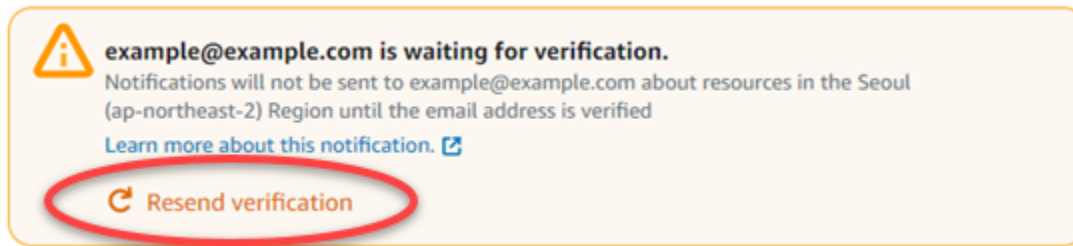
The following list describes common notification contact issues that you may experience, along with what causes them, and how to resolve them. If you encounter an issue that's not in the list,

use the **Questions? Comments?** link at the bottom of this page to submit feedback or contact the [AWS Support Center](#).

I added my email address as a notification contact but I'm not receiving email notifications

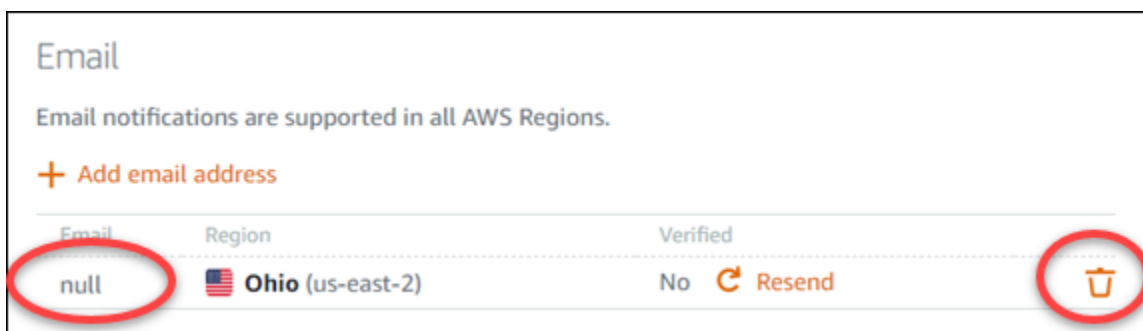
When you add an email address as a notification contact in Lightsail, a verification request is sent to that address. The verification request email contains a link that the recipient must click to confirm that they want to receive Lightsail notifications. Notifications are not sent to the email address until after it is verified. The verification comes from *AWS Notifications* <*no-reply@sns.amazonaws.com*>, with a subject of *AWS Notification - Subscription Confirmation*. SMS messaging does not require verification.

Check the mailbox's spam and junk folders if the verification request is not in the inbox folder. If the verification request got lost, or was deleted, choose **Resend verification** in the notification banner that is displayed in the Lightsail console, and in the **Account** page.



I see null listed as my email notification contact.

Email addresses must be verified within 24 hours after they are added. If you fail to verify an email within 24 hours, that email is automatically given a status of `invalid` and it is removed from Lightsail. That is why you might see a value of **null** for one or more of your email notification contacts.



To fix this issue, remove the **null** email notification contact, and add the correct email address again. Ensure that you verify the email address immediately after adding it to Lightsail. For more information, see [Notifications](#).

I have not received SMS text message notifications, or I stopped receiving them recently

You may have opted out of receiving SMS text message notifications. You can opt out by responding to an SMS text message notification with ARRET (French), CANCEL, END, OPT-OUT, OPTOUT, QUIT, REMOVE, STOP, TD, or UNSUBSCRIBE. If you opt out a mobile phone number, you must wait 30 days before you are able to add that mobile phone number again as a notification contact in Lightsail.

Troubleshoot SSL/TLS certificates in Lightsail

You might encounter errors with your Lightsail load balancers. This topic identifies common issues and workarounds for those errors.

Choose the issue below that best describes your problem, and follow the links to fix the issue. If you encounter an issue that's not in the list, use the **Questions? Comments?** link at the bottom of this page to submit feedback or contact AWS Customer Support.

I can't create a certificate.

There is a quota to the number of certificates you can create in an AWS account. For more information, see [Quotas](#) in the AWS Certificate Manager User Guide. The same quotas apply to Lightsail certificates for load balancers.

Actual error message: **Sorry, you've requested too many certificates for your account.**

My certificate request failed.

If your certificate request failed, you can **Retry** on the **Inbound traffic** tab of the load balancer management page.

If you still can't figure out what went wrong, contact AWS Customer Support.

My domain showed as invalid.

If you're having trouble verifying that you control a domain, check to see that you have access to the DNS management. If you do and you followed [these instructions](#) but still can't validate, contact AWS Customer Support.

Amazon Lightsail tutorials

The following tutorials guide you through common Amazon Lightsail use cases. For example, these tutorials show you how to troubleshoot Lightsail, and use Lightsail with other AWS services. In addition, you can learn how to work with the different Lightsail blueprints, such as Bitnami WordPress and LAMP, or Windows Server.

Topics

- [Quick start guides for Amazon Lightsail](#)
- [Bitnami tutorials for Amazon Lightsail](#)
- [WordPress tutorials for Amazon Lightsail](#)
- [WordPress Multisite tutorials for Amazon Lightsail](#)
- [Let's Encrypt tutorials for Amazon Lightsail](#)
- [Networking tutorials for Amazon Lightsail](#)
- [Work with Amazon Lightsail](#)

Quick start guides for Amazon Lightsail

Use the following quick start guides to get started with Lightsail blueprints. In Lightsail, a blueprint is a virtual image that comes prepackaged with an operating system and application. Applications include WordPress, WordPress Multisite, cPanel & WHM, PrestaShop, Drupal, Ghost, Joomla!, Magento, Redmine, LAMP, Nginx (LEMP), and Node.js

Topics

- [Quick start guide: cPanel & WHM](#)
- [Quick start guide: Drupal](#)
- [Quick start guide: Ghost](#)
- [Quick start guide: GitLab CE](#)
- [Quick start guide: Joomla!](#)
- [Quick start guide: LAMP](#)
- [Quick start guide: Magento](#)
- [Quick start guide: Nginx](#)
- [Quick start guide: Node.js](#)

- [Quick start guide: Plesk](#)
- [Quick start guide: PrestaShop](#)
- [Quick start guide: Redmine](#)
- [Quick start guide: WordPress](#)
- [Quick start guide: WordPress Multisite](#)

Quick start guide: cPanel & WHM

Here are a few steps you should take to get started after your cPanel & WHM instance is up and running on Amazon Lightsail.

Important

Your cPanel & WHM instance includes a 15-day trial license. After 15 days, you must purchase a license from cPanel to continue using cPanel & WHM. If you plan on purchasing a license, complete steps 1-7 of this guide before purchasing your license.

Contents

- [Step 1: Change the root user password](#)
- [Step 2: Attach a static IP address to your cPanel & WHM instance](#)
- [Step 3: Sign in to the Web Host Manager for the first time](#)
- [Step 4: Change the hostname and IP address of your cPanel & WHM instance](#)
- [Step 5: Map your domain name to your cPanel & WHM instance](#)
- [Step 6: Edit the firewall of your instance](#)
- [Step 7: Remove SMTP restrictions from your Lightsail instance](#)
- [Step 8: Read the cPanel & WHM documentation and get support](#)
- [Step 9: Purchase a license for cPanel & WHM](#)
- [Step 10: Create a snapshot of your cPanel & WHM instance](#)


Step 1: Change the root user password

Complete the following procedure to change the root user password on your cPanel instance. You will use the root user and password to sign in to the Web Host Manager (WHM) console later.

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.
2. After you're connected, enter the following command to change the password for the root user:

```
sudo passwd
```

3. Enter a strong password and confirm it by entering it a second time.

 **Note**

Your password should not include dictionary words and should be greater than 7 characters. If you don't follow these guidelines, you'll get a BAD PASSWORD warning.

Remember this password because you will use it to sign in to the WHM console later in this guide.

Step 2: Attach a static IP address to your cPanel & WHM instance

The default dynamic public IP address attached to your instance changes every time you stop and start the instance. Create a static IP address, and attach it to your instance, to keep the public IP address from changing. Later, when you use your domain name with your instance, you don't have to update your domain's DNS records each time you stop and start the instance. Or if your instance fails, you can restore your instance from a backup and reassign your static IP to your new instance. You can attach one static IP to an instance.

 **Important**

You must specify the public IP address of your cPanel & WHM instance when purchasing a license from cPanel. The license that you purchase is associated to that IP address. Because of this, you must attach a static IP to your cPanel & WHM instance if you plan on purchasing a license from cPanel. Specify your static IP when you purchase a license from cPanel, and keep your static IP for as long as you plan to use your cPanel & WHM license with a Lightsail instance. If you need to transfer your license to another IP address later, you can submit a request to cPanel. For more information, see [Transfer a license](#) in the *WHM documentation*.

On your instance management page, under the **Networking** tab, choose **Create static IP**, then follow the instructions on the page.

For more information, see [Create a static IP and attach it to an instance](#).

Step 3: Sign in to the Web Host Manager for the first time

Complete the following procedure to sign in to the WHM console for the first time.

1. Open a web browser and navigate to the following web address. Replace *<StaticIP>* with the static IP address of your instance. Be sure to add `:2087` to the end of the address, which is the port on which you will establish a connection to your instance.

```
https://<StaticIP>:2087
```

Example:

```
https://192.0.2.0:2087
```

Important

You must include `https://` in your browser's address bar when navigating to the IP address and port of your instance. Otherwise, you will get an error stating that the site can't be reached.

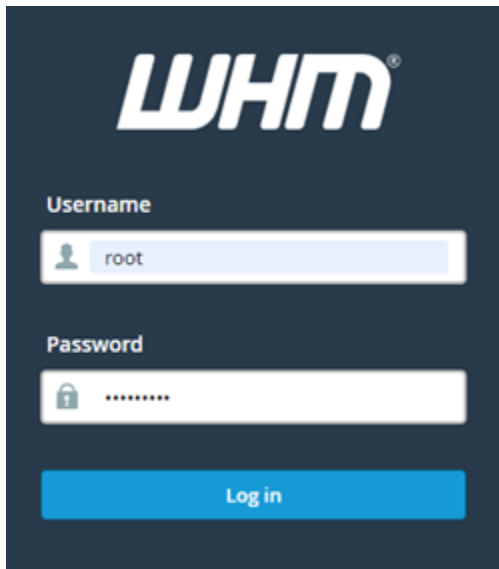
If you're unable to establish a connection when browsing to the static IP address of your instance over port 2087, check that your router, VPN, or internet service provider allows HTTP/HTTPS connections through port 2087. If it does not, then try to connect using a different network.

You might also see a browser warning that your connection is not private, not secure, or that there's a security risk. This happens because your cPanel instance does not yet have an SSL/TLS certificate applied to it. In the browser window, choose **Advanced**, **Details**, or **More information** to view the options that are available. Then choose to proceed to the website even if it's not private or secure.

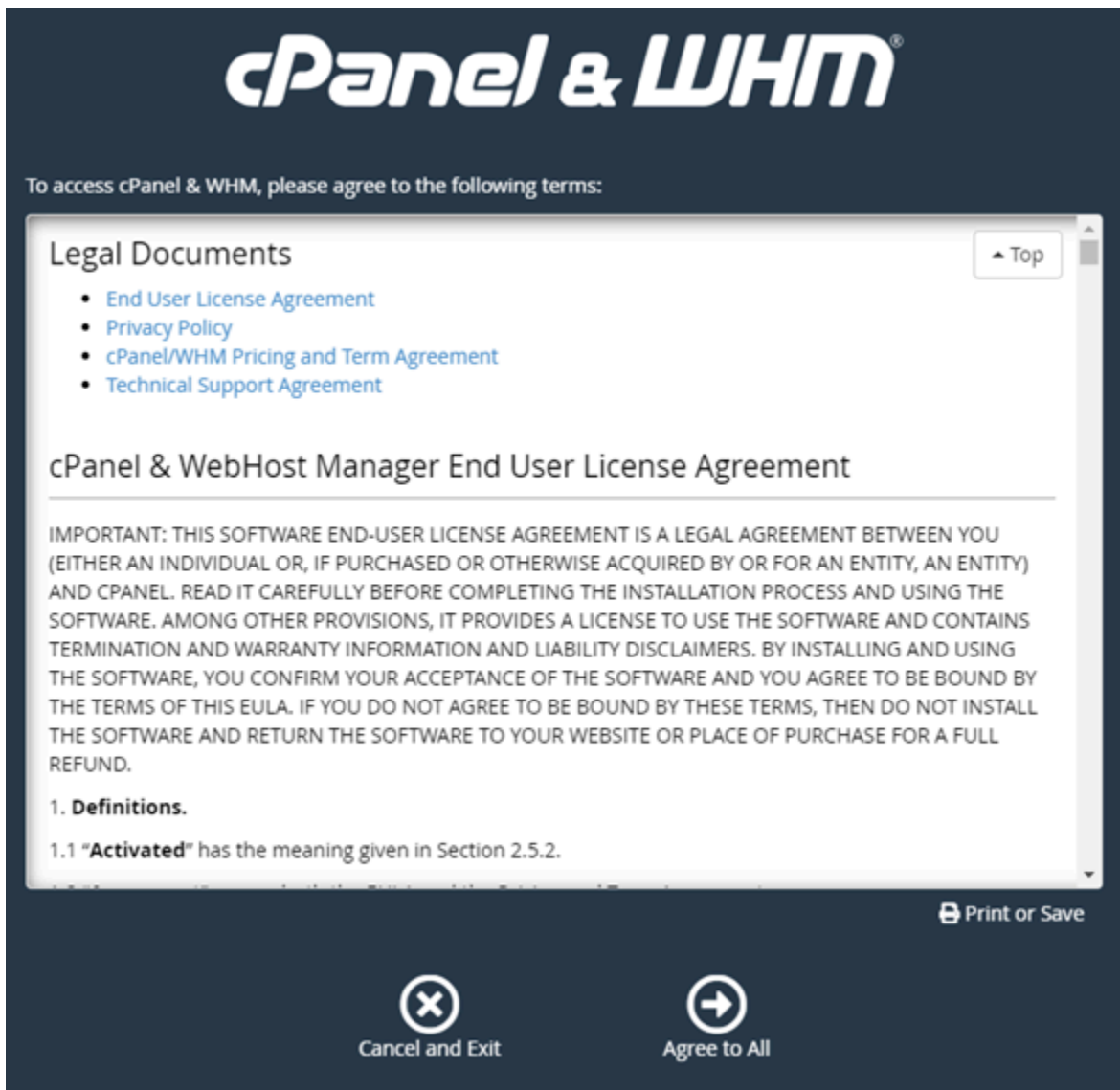
2. Enter `root` in the **Username** text box.
3. Enter the root user password in the **Password** text box.

This is the password that you specified earlier in the [Step 1: Change the root user password](#) section of this guide.

4. Choose **Log in**.

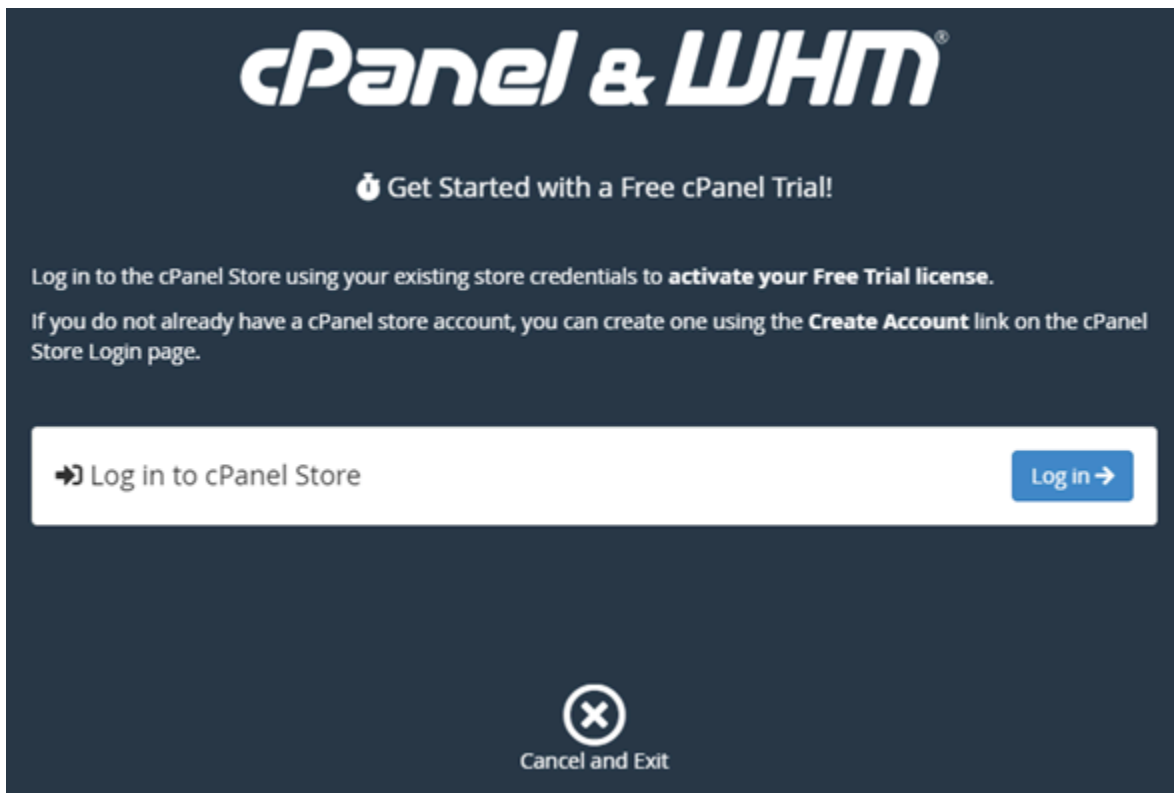
A screenshot of the WHM (Web Host Manager) login interface. The background is dark blue. At the top, the 'WHM' logo is displayed in white. Below the logo, there are two input fields. The first is labeled 'Username' and contains the text 'root'. The second is labeled 'Password' and contains a series of dots. Below the password field is a blue button with the text 'Log in' in white.

5. Read the cPanel & WHM terms, then choose **Agree to all** if you would like to proceed.



6. On the **Get started with a Free cPanel Trial** page, choose **Log in** to log in to the cPanel store.

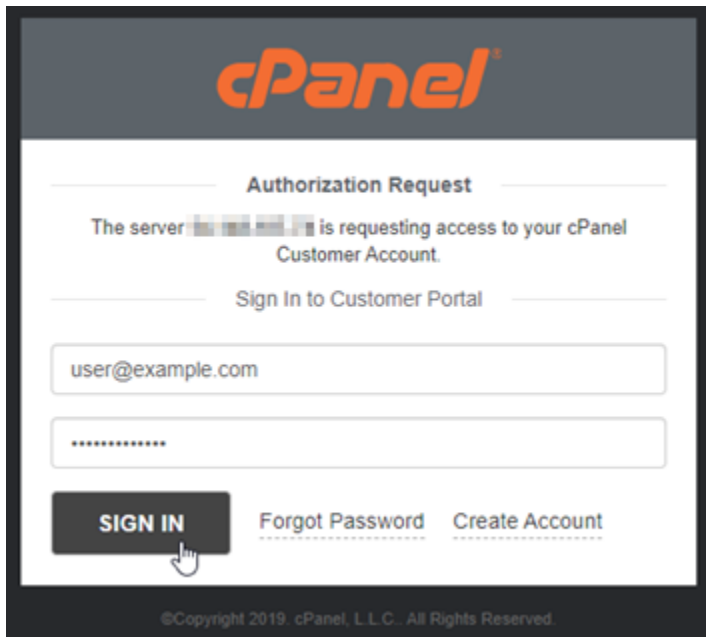
You must sign in to the cPanel store in order to associate your trial license to your account. If you don't have a cPanel store account, you should still choose **Log in**, and you will be given the option to create one.



7. In the **Authorization Request** page that appears, enter your email address or username, and the password for your cPanel store account.

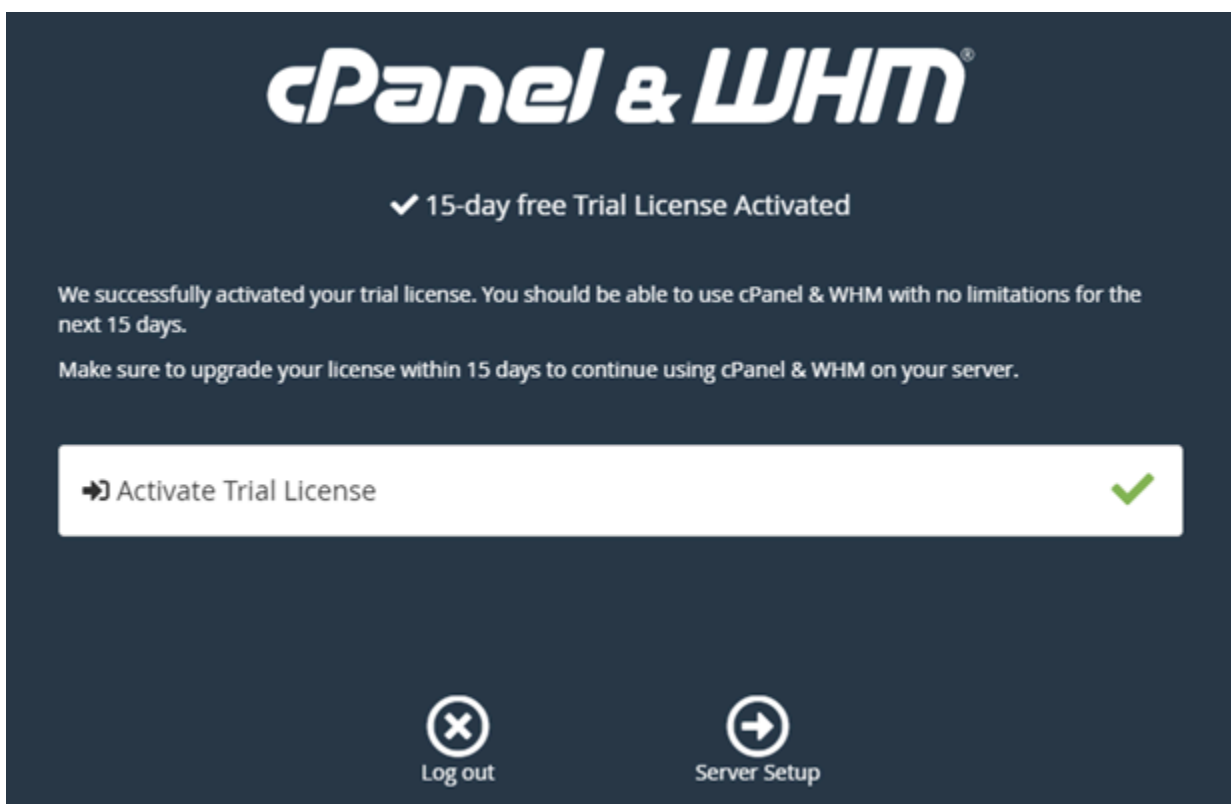
If you don't have a cPanel store account, then choose **Create Account** and follow the prompts to create your new cPanel store account. You will be asked to enter your email address, and will be sent an email to set your cPanel store account password. We recommend that you set your cPanel store account password using a new browser tab. When your password is set, you can close that tab and return to your instance to authorize your account, and continue to the next step of this procedure.

8. Choose **Sign in**.

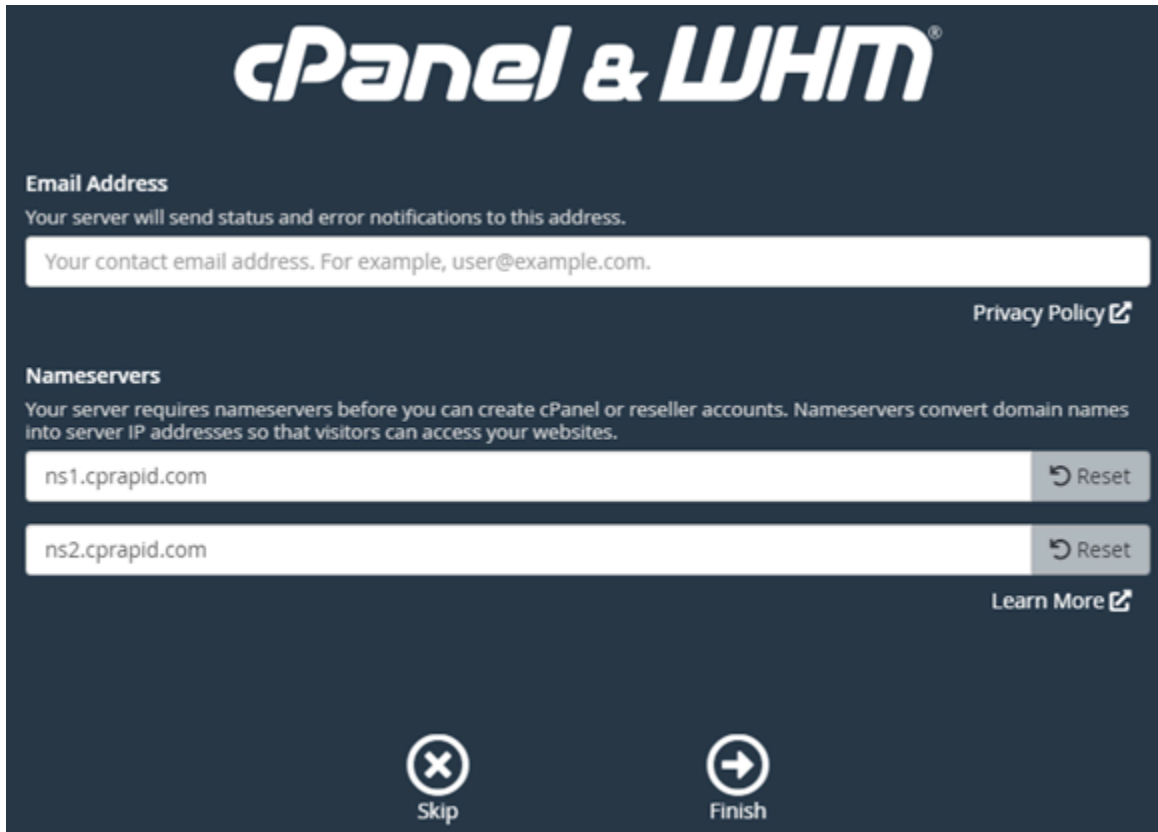


After you sign in, your cPanel & WHM instance will acquire a 15-day trial license that is associated with your cPanel store account. Go to the [Manage Licenses](#) page in the cPanel store to view your issued licenses, including trial licenses.

9. Choose **Server Setup** to continue.



10. Choose **Skip** in the email address and name servers page. You can configure these later.



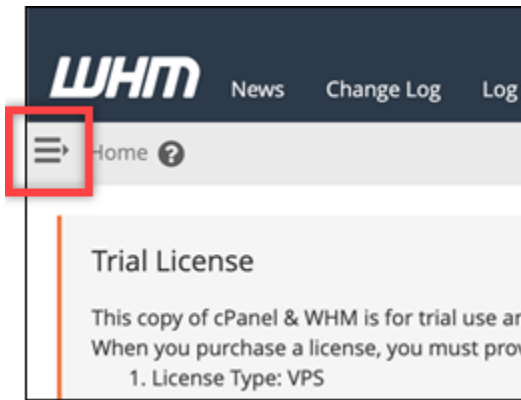
The screenshot shows the cPanel & WHM setup interface. At the top, the logo "cPanel & WHM" is displayed. Below it, the "Email Address" section has a text input field with the placeholder "Your contact email address. For example, user@example.com." and a "Privacy Policy" link. The "Nameservers" section has two text input fields, each with a "Reset" button. The first field contains "ns1.cprapid.com" and the second contains "ns2.cprapid.com". At the bottom, there are two circular buttons: "Skip" (with an 'X' icon) and "Finish" (with a right arrow icon).

The WHM console appears, where you can manage the settings and features for cPanel.

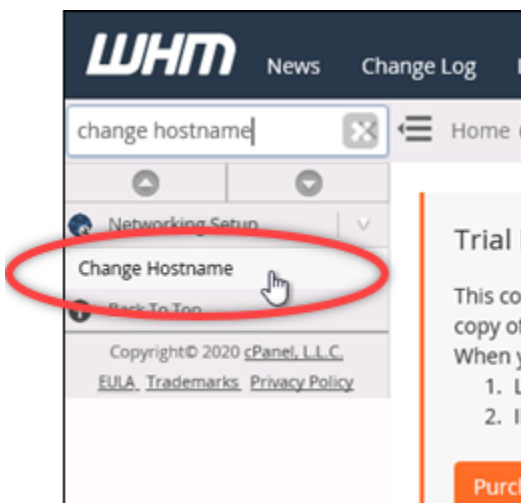
Step 4: Change the hostname and IP address of your cPanel & WHM instance

Complete the following steps to change the hostname of your instance, so that you don't have to use its public IP address to access the WHM console. You should also change the IP address of your instance to the new static IP address that you attached to your instance earlier in the [Step 2: Attach a static IP address to your cPanel & WHM instance](#) section of this guide.

1. Choose the navigation menu icon in the top-left section of the WHM console.



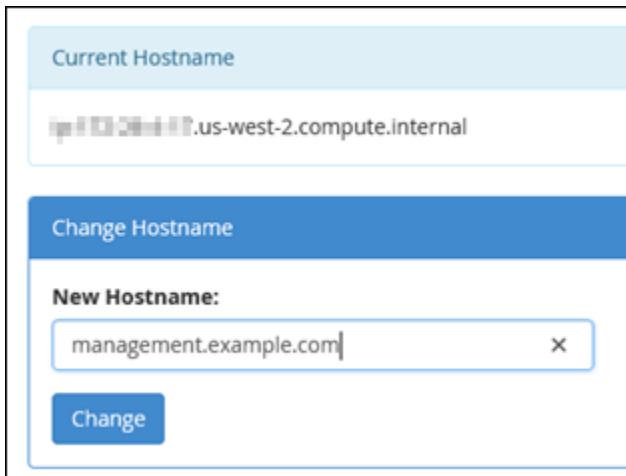
2. Enter `change hostname` in the search text box in the WHM console, then choose the **Change hostname** option in the results.



3. Enter the hostname that you want to use to access the WHM console in the **New hostname** text box. For example, enter `management.example.com` or `administration.example.com`.

Note

You can only specify a subdomain as the hostname, and you cannot specify `whm` or `cpanel` as the subdomain.



Current Hostname

ip-103-201-117.us-west-2.compute.internal

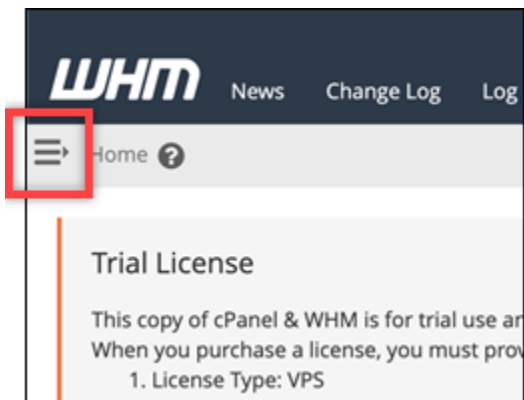
Change Hostname

New Hostname:

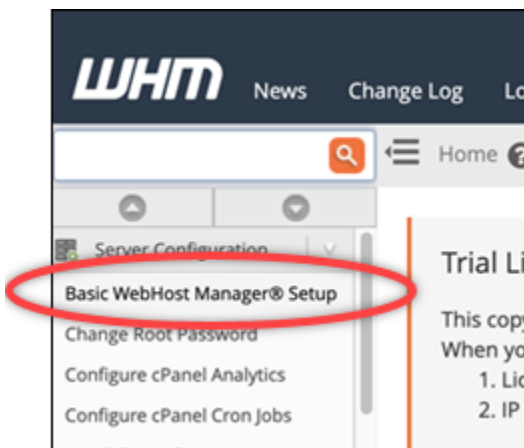
management.example.com

Change

4. Choose **Change**.
5. Choose the navigation menu icon in the top-left section of the WHM console.

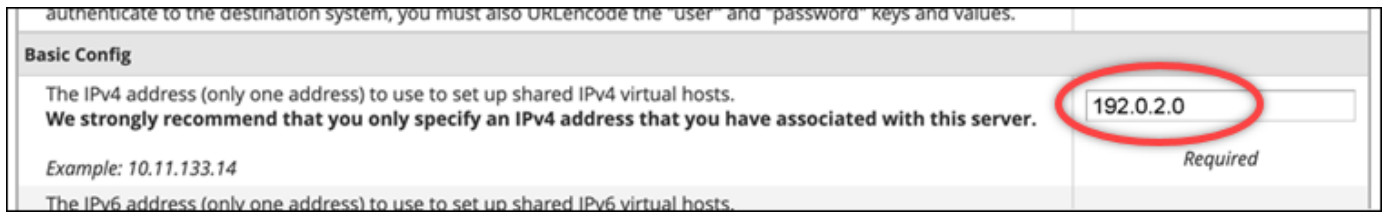


6. Choose **Basic WebHost Manager Setup**.



7. Under the **All** tab, scroll down and find the **Basic Config** section of the page.

- In the IPv4 address text box, enter the new static IP address of the instance. For information about IPv6, see [Configuring IPv6 on cPanel instances](#).



authenticate to the destination system, you must also URLencode the "user" and "password" keys and values.

Basic Config

The IPv4 address (only one address) to use to set up shared IPv4 virtual hosts.
We strongly recommend that you only specify an IPv4 address that you have associated with this server.

Example: 10.11.133.14

The IPv6 address (only one address) to use to set up shared IPv6 virtual hosts.

192.0.2.0

Required

- Scroll to the bottom of the page and choose **Save Changes**.

Note

If you receive an *Invalid License file* error message, wait and try to change the IP address again after a few minutes.

The hostname and IP address of your instance are now changed, but you must still map your domain name to your cPanel & WHM instance. You do this by adding an address (A) record in the domain name system (DNS) of your registered domain name. The A record resolves the hostname of your instance to the static IP address of your instance. We show you how to do this in the next section of this guide.

Step 5: Map your domain name to your cPanel & WHM instance

Note

You can map a domain to your cPanel & WHM instance, which you can use to access the WHM console. You can also map multiple domains within WHM, which you can use to manage websites within WHM. This section describes how to map your domain to your cPanel & WHM instance. For more information about mapping multiple domains within the WHM console, which you do when you create a new account, see [Create a new account](#) in the *WHM documentation*.

To map your domain name, such as `management.example.com` or `administration.example.com` to your instance, you add an address (A) record to the DNS of your domain. The record maps the hostname of your cPanel & WHM instance to the static IP address of your instance. The subdomain that you specify in the A record must match the hostname that you specified in the [Step 4: Change the hostname and IP address of your cPanel &](#)

[WHM instance](#) section earlier in this guide. After the A record is added, you can use the following address to access the WHM console of your instance, instead of using your instance's static IP address. Replace `<InstanceHostName>` with the host name of your instance.

```
https://<InstanceHostName>/whm
```

Example:

```
https://management.example.com/whm
```

DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console. To do this, sign in to the Lightsail console. On the Lightsail console home page, choose the **Domains & DNS** tab, and then choose **Create DNS zone**. Follow the instructions on the page to add your domain name to Lightsail. For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

Step 6: Edit the firewall of your instance

The following firewall ports are open by default on your cPanel & WHM instance:

- SSH - TCP - 22
- DNS (UDP) - UDP - 53
- DNS (TCP) - TCP - 53
- HTTP - TCP - 80
- HTTPS - TCP - 443
- Custom - TCP - 2078
- Custom - TCP - 2083
- Custom - TCP - 2087
- Custom - TCP - 2089

You might need to open additional ports depending on the services and applications that you plan to use on your instance. For example, open ports 25, 143, 465, 587, 993, 995, 2096 for email services, and ports 2080, 2091 for calendar services. Under the **Networking** tab of your instance's management page, scroll to the Firewall section of the page, and choose **Add rule**. Choose the application, protocol, and port or port range to open. Choose **Create** when you're done.

For more information about which ports to open, see [How to configure your firewall for cPanel services](#) in the *cPanel documentation*. For more information about editing your instance's firewall in Lightsail, see [Adding and editing instance firewall rules in Amazon Lightsail](#).

Step 7: Remove SMTP restrictions from your Lightsail instance

AWS blocks outbound traffic on port 25 on all Lightsail instances. To send outbound traffic on port 25, request that this restriction be removed. For more information, see [How do I remove the restriction on port 25 from my Lightsail instance?](#)

Important

If you configure SMTP to use ports 25, 465, or 587, then you must open those ports in the firewall of your instance in the Lightsail console. For more information, see [Adding and editing instance firewall rules in Amazon Lightsail](#).

Step 8: Read the cPanel & WHM documentation and get support

Read the cPanel & WHM documentation to learn how to administer web sites using cPanel and WHM. For more information, see [cPanel & WHM documentation](#).

If you have questions about cPanel & WHM or need support, you can contact cPanel using the following resources:

- [cPanel Troubleshoot your installation](#)
- [cPanel Discord channel](#)

Step 9: Purchase a license for cPanel & WHM

Your cPanel & WHM instance includes a 15-day trial license. After 15 days, you must purchase a license from cPanel to continue using cPanel & WHM. For more information, see [How to purchase a cPanel license](#) in the cPanel documentation.

Important

You must specify the public IP address of your cPanel & WHM instance when purchasing a license from cPanel. The license that you purchase is associated to that IP address. Because of this, you must attach a static IP to your cPanel & WHM instance as described in [Step](#)

[2: Attach a static IP address to your cPanel & WHM instance](#) section of this guide. Specify your static IP when you purchase a license from cPanel, and keep your static IP for as long as you plan to use your cPanel & WHM license with a Lightsail instance. If you need to transfer your license to another IP address later, you can submit a request to cPanel. For more information, see [Transfer a license](#) in the *WHM documentation*.

Step 10: Create a snapshot of your cPanel & WHM instance

A snapshot is a copy of the system disk and original configuration of an instance. A snapshot contains all of the data that is needed to restore your instance (from the moment when the snapshot was taken). You can use a snapshot as a baseline for new instances, or as a data backup. You can create a manual snapshot at any time, or you can enable automatic snapshots to have Lightsail create a daily snapshot for you.

Note

- Instance snapshots of the current generation blueprint **cPanel & WHM for AlmaLinux** can be exported to Amazon EC2.
- Instance snapshots of the previous generation blueprint **cPanel & WHM for Linux** cannot be exported to Amazon EC2 at this time.
- If you create a new instance from the snapshot, give the instance extra time to fully start up before signing into the WHM as described in [Step 3](#).

Under the **Snapshot** tab of your instance's management page, enter a name for the snapshot, then choose **Create snapshot**. Or scroll to the **Automatic snapshots** section of the page, and choose the toggle to enable automatic snapshots.

For more information, see [Create a snapshot of your Linux or Unix instance](#) and [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

Quick start guide: Drupal

Here are a few steps you should take to get started after your Drupal instance is up and running on Amazon Lightsail:

Contents

- [Step 1: Read the Bitnami documentation](#)
- [Step 2: Get the default application password to access the Drupal administration dashboard](#)
- [Step 3: Attach a static IP address to your instance](#)
- [Step 4: Sign in to the administration dashboard of your Drupal website](#)
- [Step 5: Route traffic for your registered domain name to your Drupal website](#)
- [Step 6: Configure HTTPS for your Drupal website](#)
- [Step 7: Read the Drupal documentation and continue configuring your website](#)
- [Step 8: Create a snapshot of your instance](#)

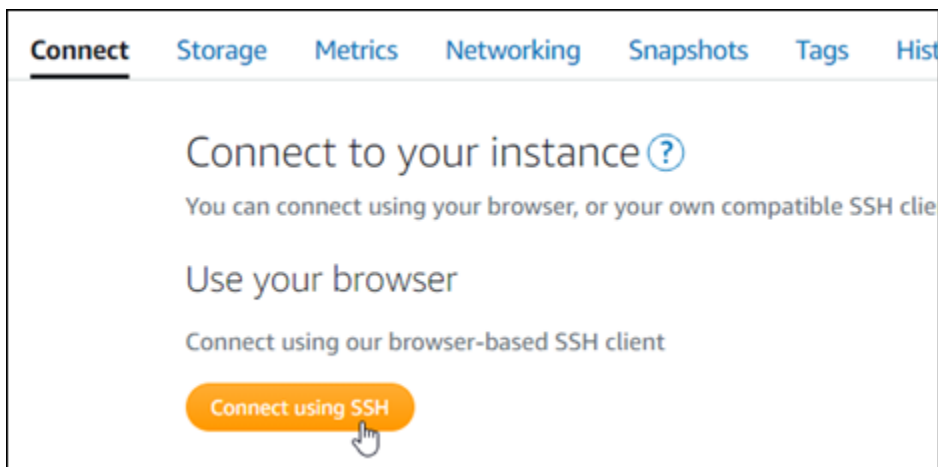
Step 1: Read the Bitnami documentation

Read the Bitnami documentation to learn how to configure your Drupal application. For more information, see the [Drupal Packaged By Bitnami For AWS Cloud](#).

Step 2: Get the default application password to access the Drupal administration dashboard

Complete the following procedure to get the default application password required to access the administration dashboard for your Drupal website. For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.

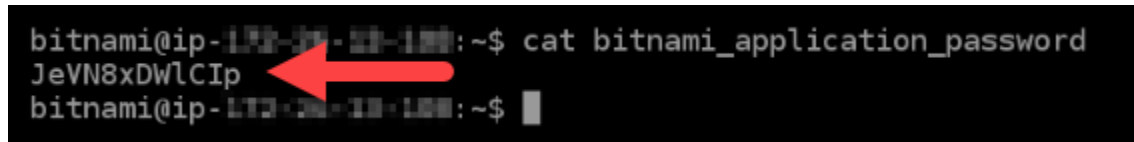


2. After you're connected, enter the following command to get the application password:

```
cat $HOME/bitnami_application_password
```

You should see a response similar to the following example, which contains the default application password:

```
bitnami@ip-192-0-2-0:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-192-0-2-0:~$
```



Step 3: Attach a static IP address to your instance

The public IP address assigned to your instance when you first create it will change every time you stop and start your instance. You should create and attach a static IP address to your instance to ensure its public IP address doesn't change. Later, when you use a registered domain name, such as `example.com`, with your instance, you don't have to update your domain's DNS records every time you stop and start your instance. You can attach one static IP to an instance.

On the instance management page, under the **Networking** tab, choose **Create a static IP** or **Attach static IP** (if you previously created a static IP that you can attach to your instance), then follow the instructions on the page. For more information, see [Create a static IP and attach it to an instance](#).

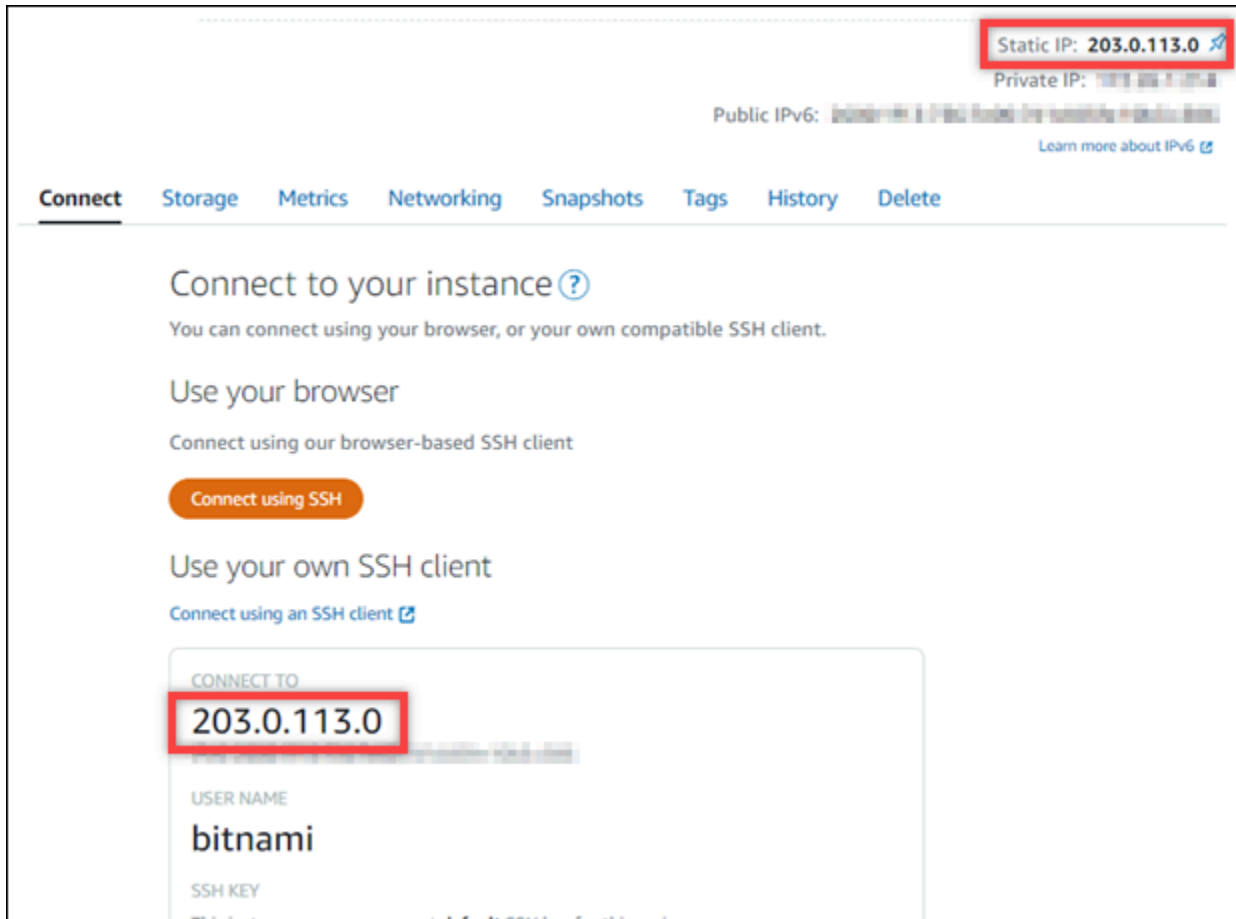


Step 4: Sign in to the administration dashboard of your Drupal website

Now that you have the default user password, navigate to your Drupal website's home page, and sign in to the administration dashboard. After you're signed in, you can start customizing your website and making administrative changes. For more information about what you can do in

Drupal, see the [Step 7: Read the Drupal documentation and continue configuring your website](#) section later in this guide.

1. On your instance management page, under the **Connect** tab, make note of the public IP address of your instance. The public IP address is also displayed in the header section of your instance management page.



2. Browse to the public IP address of your instance, for example by going to `http://203.0.113.0`.

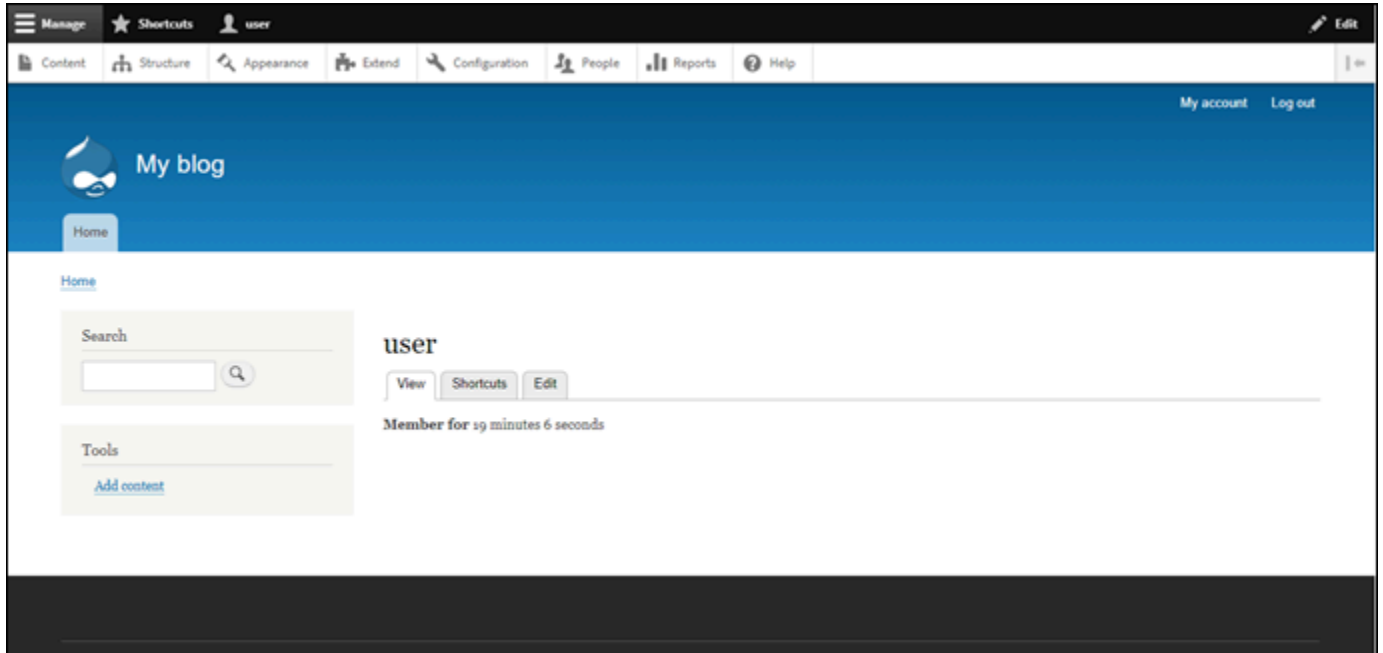
The home page of your Drupal website should appear.

3. Choose **Manage** in the bottom right corner of your Drupal website home page.

If the **Manage** banner is not shown, you can reach the sign in page by browsing to `http://<PublicIP>/user/login`. Replace `<PublicIP>` with the public IP address of your instance.

4. Sign in using the default user name (`user`) and the default password retrieved earlier in this guide.

The Drupal administration dashboard appears.



Step 5: Route traffic for your registered domain name to your Drupal website

To route traffic for your registered domain name, such as `example.com`, to your Drupal website, you add a record to the domain name system (DNS) of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page. For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

If you browse to the domain name that you configured for your instance, you should be redirected to the home page of your Drupal website. Next, you should generate and configure an SSL/TLS certificate to enable HTTPS connections for your Drupal website. For more information, continue to the next [Step 6: Configure HTTPS for your Drupal website](#) section of this guide.

Step 6: Configure HTTPS for your Drupal website

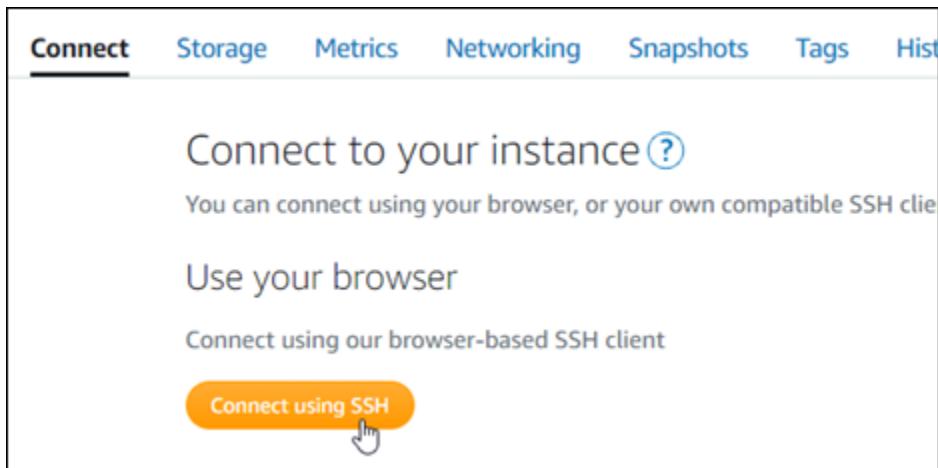
Complete the following procedure to configure HTTPS on your Drupal website. These steps show you how to use the Bitnami HTTPS Configuration Tool (`bncert-tool`), which is a command line

tool for requesting Let's Encrypt SSL/TLS certificates. For more information see [Learn About The Bitnami HTTPS Configuration Tool](#) in the *Bitnami documentation*.

⚠ Important

Before starting with this procedure, make sure that you configured your domain to route traffic to your Drupal instance. Otherwise, the SSL/TLS certificate validation process will fail.

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to confirm the bncert tool is installed on your instance.

```
sudo /opt/bitnami/bncert-tool
```

You should see one of the following responses:

- If you see command not found in the response, then the bncert tool is not installed on your instance. Continue to the next step in this procedure to install the bncert tool on your instance.
- If you see **Welcome to the Bitnami HTTPS configuration tool** in the response, then the bncert tool is installed on your instance. Continue to the step 8 of this procedure.
- If the bncert tool has been installed on your instance for a while, then you might see a message indicating that an updated version of the tool is available. Choose to download it, and then enter the `sudo /opt/bitnami/bncert-tool` command to run the bncert tool again. Continue to the step 8 of this procedure.

3. Enter the following command to download the bncert run file to your instance.

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/bncert-linux-x64.run
```

4. Enter the following command to create a directory for the bncert tool run file on your instance.

```
sudo mkdir /opt/bitnami/bncert
```

5. Enter the following command to make the bncert run a file that can be executed as a program.

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. Enter the following command to create a symbolic link that runs the bncert tool when you enter the `sudo /opt/bitnami/bncert-tool` command.

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

You are now done installing the bncert tool on your instance.

7. Enter the following command to run the bncert tool.

```
sudo /opt/bitnami/bncert-tool
```

8. Enter your primary domain name and alternate domain names separated by a space as shown in the following example.

If your domain is not configured to route traffic to the public IP address of your instance, the bncert tool will ask you to make that configuration before continuing. Your domain must be routing traffic to the public IP address of the instance from which you are using the bncert tool to enable HTTPS on the instance. This confirms that you own the domain, and serves as the validation for your certificate.

```
-----
Welcome to the Bitnami HTTPS Configuration tool.
-----
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: example.com www.example.com
```

9. The `bncert` tool will ask you how you want your website's redirection to be configured. These are the options available:
- **Enable HTTP to HTTPS redirection** - Specifies whether users who browse to the HTTP version of your website (i.e., `http://example.com`) are automatically redirected to the HTTPS version (i.e., `https://example.com`). We recommend enabling this option because it forces all visitors to use the encrypted connection. Type **Y** and press **Enter** to enable it.
 - **Enable non-www to www redirection** - Specifies whether users who browse to the apex of your domain (i.e., `https://example.com`) are automatically redirected to your domain's `www` subdomain (i.e., `https://www.example.com`). We recommend enabling this option. However, you may want to disable it and enable the alternate option (enable `www` to non-`www` redirection) if you have specified the apex of your domain as your preferred website address in search engine tools like Google's webmaster tools, or if your apex points directly to your IP and your `www` subdomain references your apex via a CNAME record. Type **Y** and press **Enter** to enable it.
 - **Enable www to non-www redirection** - Specifies whether users who browse to your domain's `www` subdomain (i.e., `https://www.example.com`) are automatically redirected to the apex of your domain (i.e., `https://example.com`). We recommend disabling this, if you enabled non-`www` redirection to `www`. Type **N** and press **Enter** to disable it.

Your selections should look like the following example.

```
Enable/disable redirections
Please select the redirections you wish to enable or disable on your Bitnami
installation.

Enable HTTP to HTTPS redirection [Y/n]: Y

Enable non-www to www redirection [Y/n]: Y

Enable www to non-www redirection [y/N]: N
```

10. The changes that are going to be made are listed. Type **Y** and press **Enter** to confirm and continue.

```

Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
www.example.com)
7. Start web server once all changes have been performed

Do you agree to these changes? [Y/n]: Y

```

11. Enter your email address to associate with your Let's Encrypt certificate and press **Enter**.

```

Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []:

```

12. Review the Let's Encrypt Subscriber Agreement. Type Y and press **Enter** to accept the agreement and continue.

```

The Let's Encrypt Subscriber Agreement can be found at:

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]:

```

The actions are performed to enable HTTPS on your instance, including requesting the certificate and configuring the redirections you specified.

```

Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.

|

```

Your certificate is successfully issued and validated, and the redirections are successfully configured on your instance if you see a message similar to the following example.

```
Success

The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.

The configuration report is shown below.

Backup files:
* /opt/bitnami/apache2/conf/httpd.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami.conf.back.202005290035

Find more details in the log file:

/tmp/bncert-202005290035.log

If you find any issues, please check Bitnami Support forums at:

https://community.bitnami.com

Press [Enter] to continue:█
```

The bncert tool will perform an automatic renewal of your certificate every 80 days before it expires. Repeat the above steps if you wish to use additional domains and subdomains with your instance, and you want to enable HTTPS for those domains.

You are now done enabling HTTPS on your Drupal instance. Next time you browse to your Drupal website using the domain you configured, you should see that it redirects to the HTTPS connection.

Step 7: Read the Drupal documentation and continue configuring your website

Read the Drupal documentation to learn how to administer and customize your website. For more information, see the [Drupal Documentation](#).

Step 8: Create a snapshot of your instance

After you configure your Drupal website the way you want it, create periodic snapshots of your instance to back it up. You can create snapshots manually, or enable automatic snapshots to have Lightsail create daily snapshots for you. If something goes wrong with your instance, you can create a new replacement instance using the snapshot. For more information, see [Snapshots](#).

On the instance management page, under the **Snapshot** tab, choose **Create a snapshot** or choose to enable automatic snapshots.

Connect
Storage
Metrics
Networking
Snapshots
Tags
History
Delete

Manual snapshots ?

You can create a snapshot to back up your instance, its system disk, and attached disks.

[+ Create snapshot](#)

> February 5, 2021 - 9:37 AM	"Prestashop-1612546662"	⋮
> January 13, 2021 - 9:44 AM	"Prestashop-1610559880"	⋮
> December 9, 2020 - 12:33 PM	"Prestashop-1607545986"	⋮
> September 9, 2020 - 5:44 PM	"Prestashop-1599698658"	⋮

Showing 4 of 4 snapshots

Automatic snapshots ?

Automatic snapshots are enabled

Your daily snapshot time is 10:00 PM PST.
We will store your seven most recent snapshots.

[Change snapshot time](#)

DAILY SNAPSHOTS

> Thursday	March 4, 2021	⋮
> Wednesday	March 3, 2021	⋮
> Tuesday	March 2, 2021	⋮

For more information, see [Creating a snapshot of your Linux or Unix instance in Amazon Lightsail](#) or [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

Quick start guide: Ghost

Here are a few steps you should take to get started after your Ghost instance is up and running on Amazon Lightsail:

Contents

- [Step 1: Read the Bitnami documentation](#)
- [Step 2: Get the default application password to access the Ghost administration dashboard](#)

- [Step 3: Attach a static IP address to your instance](#)
- [Step 4: Sign in to the administration dashboard of your Ghost website](#)
- [Step 5: Route traffic for your registered domain name to your Ghost website](#)
- [Step 6: Configure HTTPS for your Ghost website](#)
- [Step 7: Read the Ghost documentation and continue configuring your website](#)
- [Step 8: Create a snapshot of your instance](#)

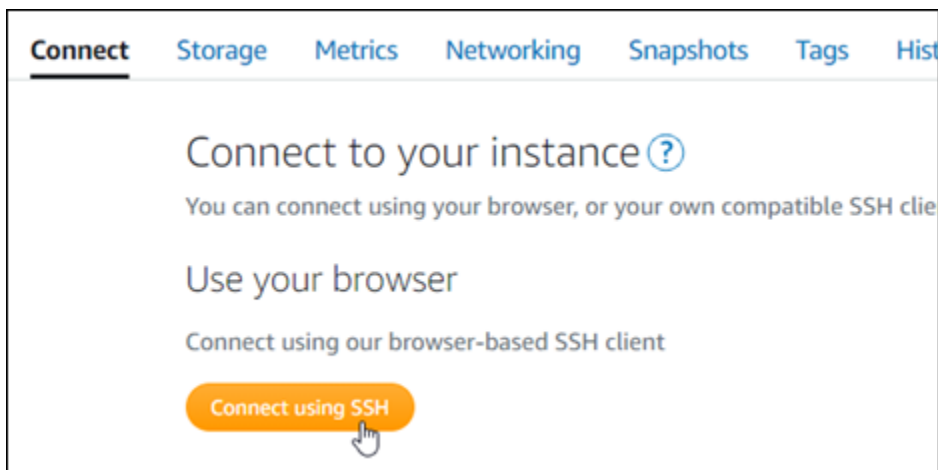
Step 1: Read the Bitnami documentation

Read the Bitnami documentation to learn how to configure your Ghost application. For more information, see the [Ghost Packaged By Bitnami For AWS Cloud](#).

Step 2: Get the default application password to access the Ghost administration dashboard

Complete the following procedure to get the default application password required to access the administration dashboard for your Ghost website. For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.

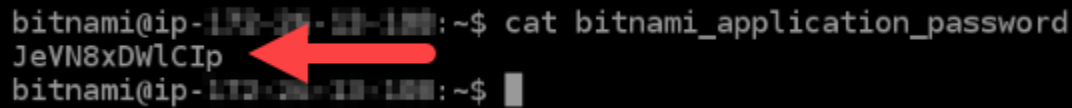


2. After you're connected, enter the following command to get the application password:

```
cat $HOME/bitnami_application_password
```

You should see a response similar to the following example, which contains the default application password:

```
bitnami@ip-192-0-2-0:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-192-0-2-0:~$
```



Step 3: Attach a static IP address to your instance

The public IP address assigned to your instance when you first create it will change every time you stop and start your instance. You should create and attach a static IP address to your instance to ensure its public IP address doesn't change. Later, when you use a registered domain name, such as `example.com`, with your instance, you don't have to update your domain's DNS records every time you stop and start your instance. You can attach one static IP to an instance.

On the instance management page, under the **Networking** tab, choose **Create a static IP** or **Attach static IP** (if you previously created a static IP that you can attach to your instance), then follow the instructions on the page. For more information, see [Create a static IP and attach it to an instance](#).

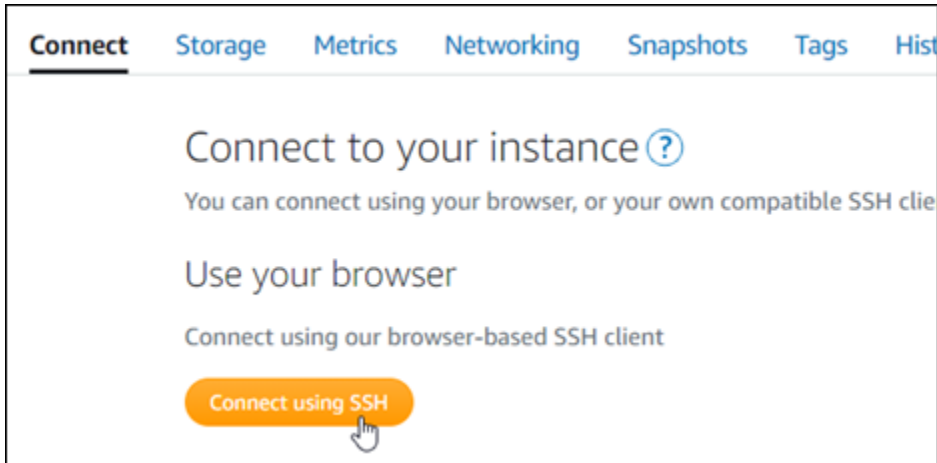


After the new static IP address is attached to your instance, you must complete the following steps to make the application aware of the new static IP address.

1. Make a note of the static IP address of your instance. It's listed in the header section of your instance management page.



2. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



3. After you're connected, enter the following command. Replace *<StaticIP>* with the new static IP address of your instance.

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

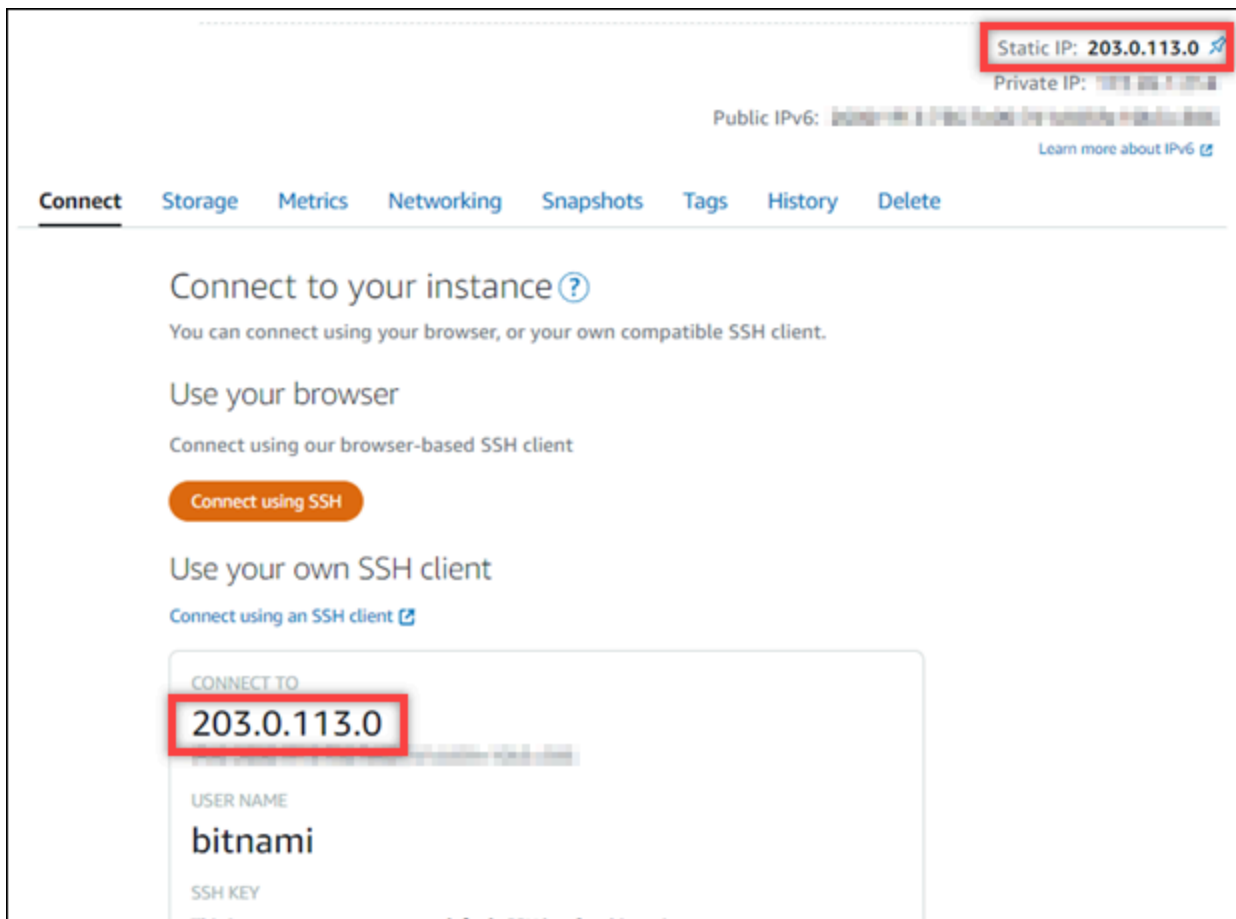
You should see a response similar to the following example. The application on your instance should now be aware of the new static IP address.

```
bitnami@ip-173-20-0-100:~$ sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
Configuring domain to 203.0.113.0
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO ==> Updating hostname in database
prestashop 15:49:22.46 INFO ==> Purging cache
Disabling automatic domain update for IP address changes
```


Step 4: Sign in to the administration dashboard of your Ghost website

Now that you have the default application password, complete the following procedure to navigate to your Ghost website's home page, and sign in to the administration dashboard. After you're signed in, you can start customizing your website and making administrative changes. For more information about what you can do in Ghost, see the [Step 6: Read the Ghost documentation and continue configuring your website](#) section later in this guide.

1. On your instance management page, under the **Connect** tab, make note of the public IP address of your instance. The public IP address is also displayed in the header section of your instance management page.



2. Browse to the public IP address of your instance, for example by going to `http://203.0.113.0`.

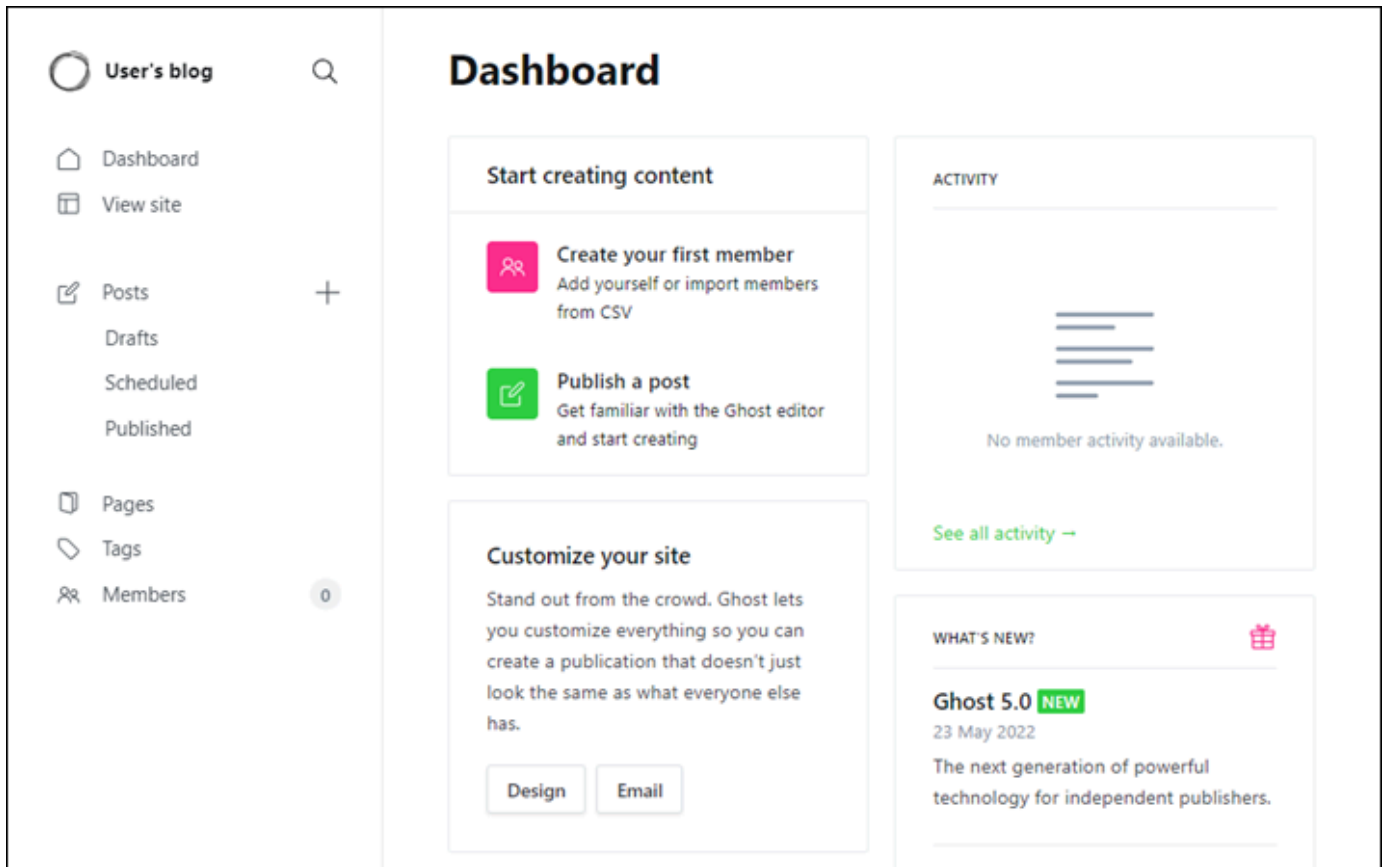
The home page of your Ghost website should appear.

3. Choose **Manage** in the bottom right corner of your Ghost website home page.

If the **Manage** banner is not shown, you can reach the sign in page by browsing to `http://<PublicIP>/ghost`. Replace `<PublicIP>` with the public IP address of your instance.

4. Sign in using the default user name (`user@example.com`) and the default password retrieved earlier in this guide.

The Ghost administration dashboard appears.



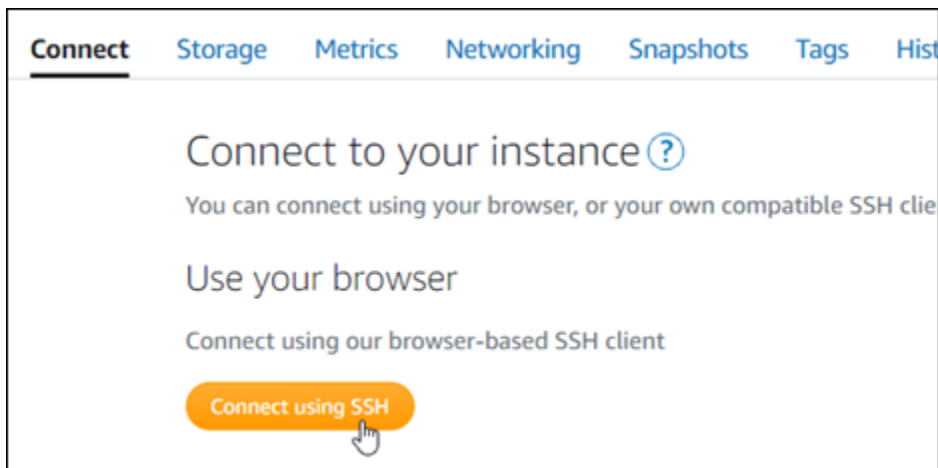
Step 5: Route traffic for your registered domain name to your Ghost website

To route traffic for your registered domain name, such as `example.com`, to your Ghost website, you add a record to the DNS of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page. For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

After your domain name is routing traffic to your instance, you must complete the following steps to make the Ghost application aware of the new domain.

1. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command. Replace *<DomainName>* with the domain name that is directing traffic to your Ghost instance.

```
sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain example.com
```

You should see a response similar to the following example. The Ghost application should now be aware of the domain.

```
bitnami@ip-10.10.10.10:~$ sudo /opt/bitnami/configure_app_domain --domain example.com
Configuring domain to example.com
2022-06-09T22:25:58.177Z - info: Saving configuration info to disk
ghost 22:25:58.57 INFO ==> Configuring Ghost URL to http://example.com
Disabling automatic domain update for IP address changes
```

If you browse to the domain name that you configured for your instance, you should be redirected to the home page of your Ghost website. Next, you should generate and configure an SSL/TLS

certificate to enable HTTPS connections for your Ghost website. For more information, continue to the next [Step 6: Configure HTTPS for your Ghost website](#) section of this guide.

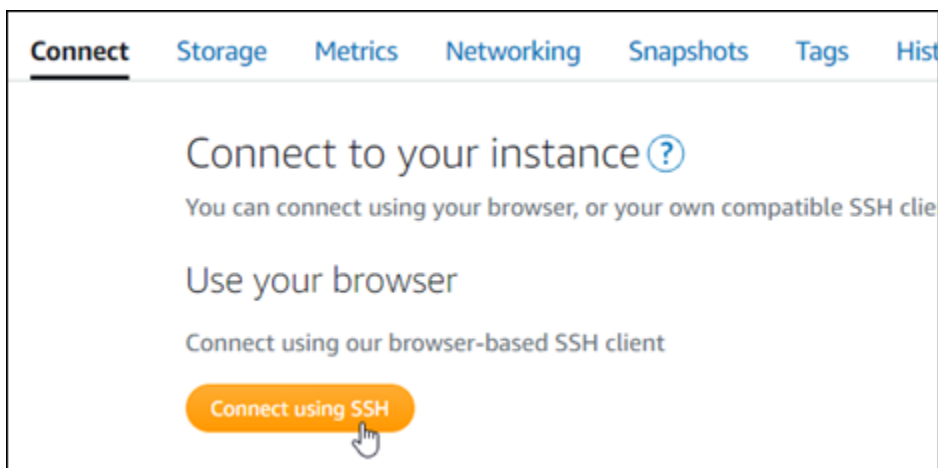
Step 6: Configure HTTPS for your Ghost website

Complete the following procedure to configure HTTPS on your Ghost website. These steps show you how to use the Bitnami HTTPS Configuration Tool (`bncert-tool`), which is a command line tool for requesting Let's Encrypt SSL/TLS certificates. For more information see [Learn About The Bitnami HTTPS Configuration Tool](#) in the *Bitnami documentation*.

Important

Before starting with this procedure, make sure that you configured your domain to route traffic to your Ghost instance. Otherwise, the SSL/TLS certificate validation process will fail.

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to confirm the `bncert` tool is installed on your instance.

```
sudo /opt/bitnami/bncert-tool
```

You should see one of the following responses:

- If you see `command not found` in the response, then the `bncert` tool is not installed on your instance. Continue to the next step in this procedure to install the `bncert` tool on your instance.

- If you see **Welcome to the Bitnami HTTPS configuration tool** in the response, then the `bncert` tool is installed on your instance. Continue to the step 8 of this procedure.
 - If the `bncert` tool has been installed on your instance for a while, then you might see a message indicating that an updated version of the tool is available. Choose to download it, and then enter the `sudo /opt/bitnami/bncert-tool` command to run the `bncert` tool again. Continue to the step 8 of this procedure.
3. Enter the following command to download the `bncert` run file to your instance.

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/bncert-linux-x64.run
```

4. Enter the following command to create a directory for the `bncert` tool run file on your instance.

```
sudo mkdir /opt/bitnami/bncert
```

5. Enter the following command to make the `bncert` run a file that can be executed as a program.

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. Enter the following command to create a symbolic link that runs the `bncert` tool when you enter the `sudo /opt/bitnami/bncert-tool` command.

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

You are now done installing the `bncert` tool on your instance.

7. Enter the following command to run the `bncert` tool.

```
sudo /opt/bitnami/bncert-tool
```

8. Enter your primary domain name and alternate domain names separated by a space as shown in the following example.

If your domain is not configured to route traffic to the public IP address of your instance, the `bncert` tool will ask you to make that configuration before continuing. Your domain must be routing traffic to the public IP address of the instance from which you are using the `bncert` tool to enable HTTPS on the instance. This confirms that you own the domain, and serves as the validation for your certificate.

```
-----  
Welcome to the Bitnami HTTPS Configuration tool.  
-----  
Domains  
Please provide a valid space-separated list of domains for which you wish to  
configure your web server.  
Domain list []: example.com www.example.com
```

9. The `bncert` tool will ask you how you want your website's redirection to be configured. These are the options available:
- **Enable HTTP to HTTPS redirection** - Specifies whether users who browse to the HTTP version of your website (i.e., `http://example.com`) are automatically redirected to the HTTPS version (i.e., `https://example.com`). We recommend enabling this option because it forces all visitors to use the encrypted connection. Type **Y** and press **Enter** to enable it.
 - **Enable non-www to www redirection** - Specifies whether users who browse to the apex of your domain (i.e., `https://example.com`) are automatically redirected to your domain's `www` subdomain (i.e., `https://www.example.com`). We recommend enabling this option. However, you may want to disable it and enable the alternate option (enable `www` to non-`www` redirection) if you have specified the apex of your domain as your preferred website address in search engine tools like Google's webmaster tools, or if your apex points directly to your IP and your `www` subdomain references your apex via a CNAME record. Type **Y** and press **Enter** to enable it.
 - **Enable www to non-www redirection** - Specifies whether users who browse to your domain's `www` subdomain (i.e., `https://www.example.com`) are automatically redirected to the apex of your domain (i.e., `https://example.com`). We recommend disabling this, if you enabled non-`www` redirection to `www`. Type **N** and press **Enter** to disable it.

Your selections should look like the following example.

```
Enable/disable redirections

Please select the redirections you wish to enable or disable on your Bitnami
installation.

Enable HTTP to HTTPS redirection [Y/n]: Y

Enable non-www to www redirection [Y/n]: Y

Enable www to non-www redirection [y/N]: N
```

10. The changes that are going to be made are listed. Type Y and press **Enter** to confirm and continue.

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
www.example.com)
7. Start web server once all changes have been performed

Do you agree to these changes? [Y/n]: Y
```

11. Enter your email address to associate with your Let's Encrypt certificate and press **Enter**.

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []: 
```

12. Review the Let's Encrypt Subscriber Agreement. Type Y and press **Enter** to accept the agreement and continue.

```
The Let's Encrypt Subscriber Agreement can be found at:  
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf  
Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]: █
```

The actions are performed to enable HTTPS on your instance, including requesting the certificate and configuring the redirections you specified.

```
Performing changes to your installation  
  
The Bitnami HTTPS Configuration Tool will perform any necessary actions to your  
Bitnami installation. This may take some time, please be patient.  
  
█
```

Your certificate is successfully issued and validated, and the redirections are successfully configured on your instance if you see a message similar to the following example.

```
Success  
  
The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.  
The configuration report is shown below.  
  
Backup files:  
* /opt/bitnami/apache2/conf/httpd.conf.back.202005290035  
* /opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf.back.202005290035  
* /opt/bitnami/apache2/conf/bitnami/bitnami.conf.back.202005290035  
  
Find more details in the log file:  
  
/tmp/bncert-202005290035.log  
  
If you find any issues, please check Bitnami Support forums at:  
  
https://community.bitnami.com  
  
Press [Enter] to continue:█
```

The bncert tool will perform an automatic renewal of your certificate every 80 days before it expires. Repeat the above steps if you wish to use additional domains and subdomains with your instance, and you want to enable HTTPS for those domains.

You are now done enabling HTTPS on your Ghost instance. Next time you browse to your Ghost website using the domain you configured, you should see that it redirects to the HTTPS connection.

Step 7: Read the Ghost documentation and continue configuring your website

Read the Ghost documentation to learn how to administer and customize your website. For more information, see the [Ghost Documentation](#).

Step 8: Create a snapshot of your instance

After you configure your Ghost website the way you want it, create periodic snapshots of your instance to back it up. You can create snapshots manually, or enable automatic snapshots to have Lightsail create daily snapshots for you. If something goes wrong with your instance, you can create a new replacement instance using the snapshot. For more information, see [Snapshots](#).

On the instance management page, under the **Snapshot** tab, choose **Create a snapshot** or choose to enable automatic snapshots.

Connect
Storage
Metrics
Networking
Snapshots
Tags
History
Delete

Manual snapshots ?

You can create a snapshot to back up your instance, its system disk, and attached disks.

[+ Create snapshot](#)

> February 5, 2021 - 9:37 AM	"Prestashop-1612546662"	⋮
> January 13, 2021 - 9:44 AM	"Prestashop-1610559880"	⋮
> December 9, 2020 - 12:33 PM	"Prestashop-1607545986"	⋮
> September 9, 2020 - 5:44 PM	"Prestashop-1599698658"	⋮

Showing 4 of 4 snapshots

Automatic snapshots ?

Automatic snapshots are enabled

Your daily snapshot time is 10:00 PM PST.
We will store your seven most recent snapshots.

[Change snapshot time](#)

DAILY SNAPSHOTS

> Thursday	March 4, 2021	⋮
> Wednesday	March 3, 2021	⋮
> Tuesday	March 2, 2021	⋮

For more information, see [Creating a snapshot of your Linux or Unix instance in Amazon Lightsail](#) or [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

Quick start guide: GitLab CE

Here are a few steps you should take to get started after your GitLab CE instance is up and running on Amazon Lightsail:

Contents

- [Step 1: Read the Bitnami documentation](#)
- [Step 2: Get the default application password to access the GitLab CE admin area](#)

- [Step 3: Attach a static IP address to your instance](#)
- [Step 4: Sign in to the admin area of your GitLab CE website](#)
- [Step 5: Route traffic for your registered domain name to your GitLab CE website](#)
- [Step 6: Configure HTTPS for your GitLab CE website](#)
- [Step 7: Read the GitLab CE documentation and continue configuring your website](#)
- [Step 8: Create a snapshot of your instance](#)

Step 1: Read the Bitnami documentation

Read the Bitnami documentation to learn how to configure your GitLab CE application. For more information, see the [GitLab CE Packaged By Bitnami For AWS Cloud](#).

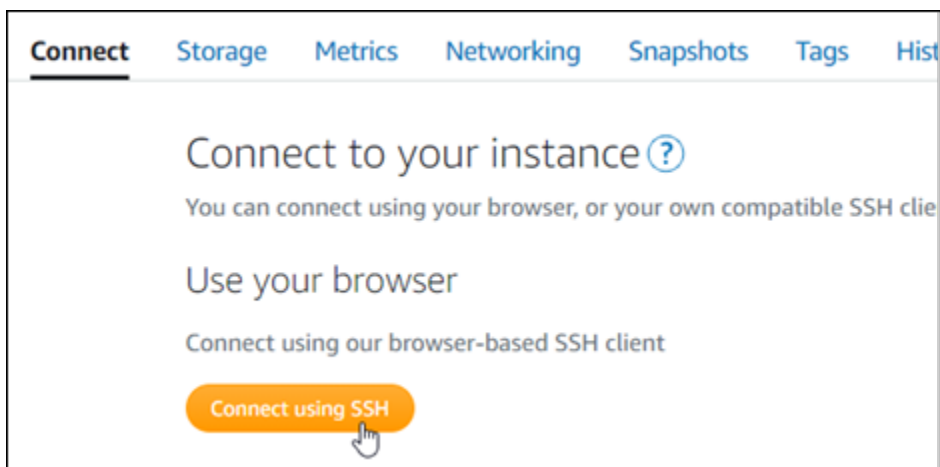
Step 2: Get the default application password to access the GitLab CE admin area

Complete the following procedure to get the default application password required to access the admin area for your GitLab CE website. For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to get the application password:

```
cat $HOME/bitnami_application_password
```

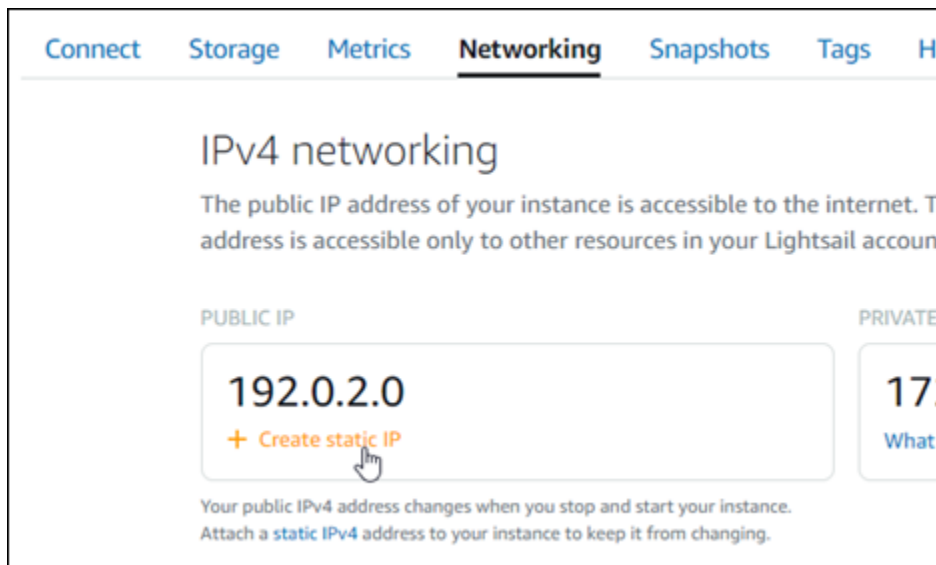
You should see a response similar to the following example, which contains the default application password:

```
bitnami@ip-192-0-2-0:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-192-0-2-0:~$
```

Step 3: Attach a static IP address to your instance

The public IP address assigned to your instance when you first create it will change every time you stop and start your instance. You should create and attach a static IP address to your instance to ensure its public IP address doesn't change. Later, when you use a registered domain name, such as `example.com`, with your instance, you don't have to update your domain's DNS records every time you stop and start your instance. You can attach one static IP to an instance.

On the instance management page, under the **Networking** tab, choose **Create a static IP** or **Attach static IP** (if you previously created a static IP that you can attach to your instance), then follow the instructions on the page. For more information, see [Create a static IP and attach it to an instance](#).

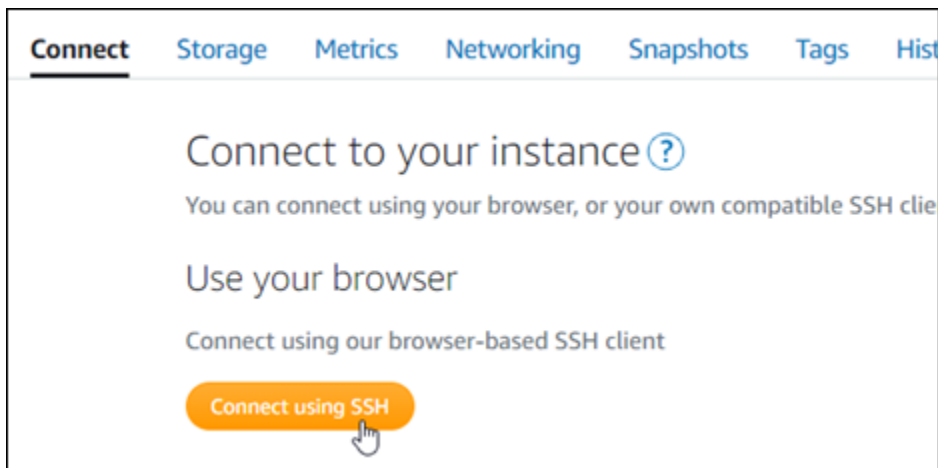


After the new static IP address is attached to your instance, you must complete the following steps to make the application aware of the new static IP address.

1. Make a note of the static IP address of your instance. It's listed in the header section of your instance management page.



2. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



3. After you're connected, enter the following command. Replace *<StaticIP>* with the new static IP address of your instance.

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

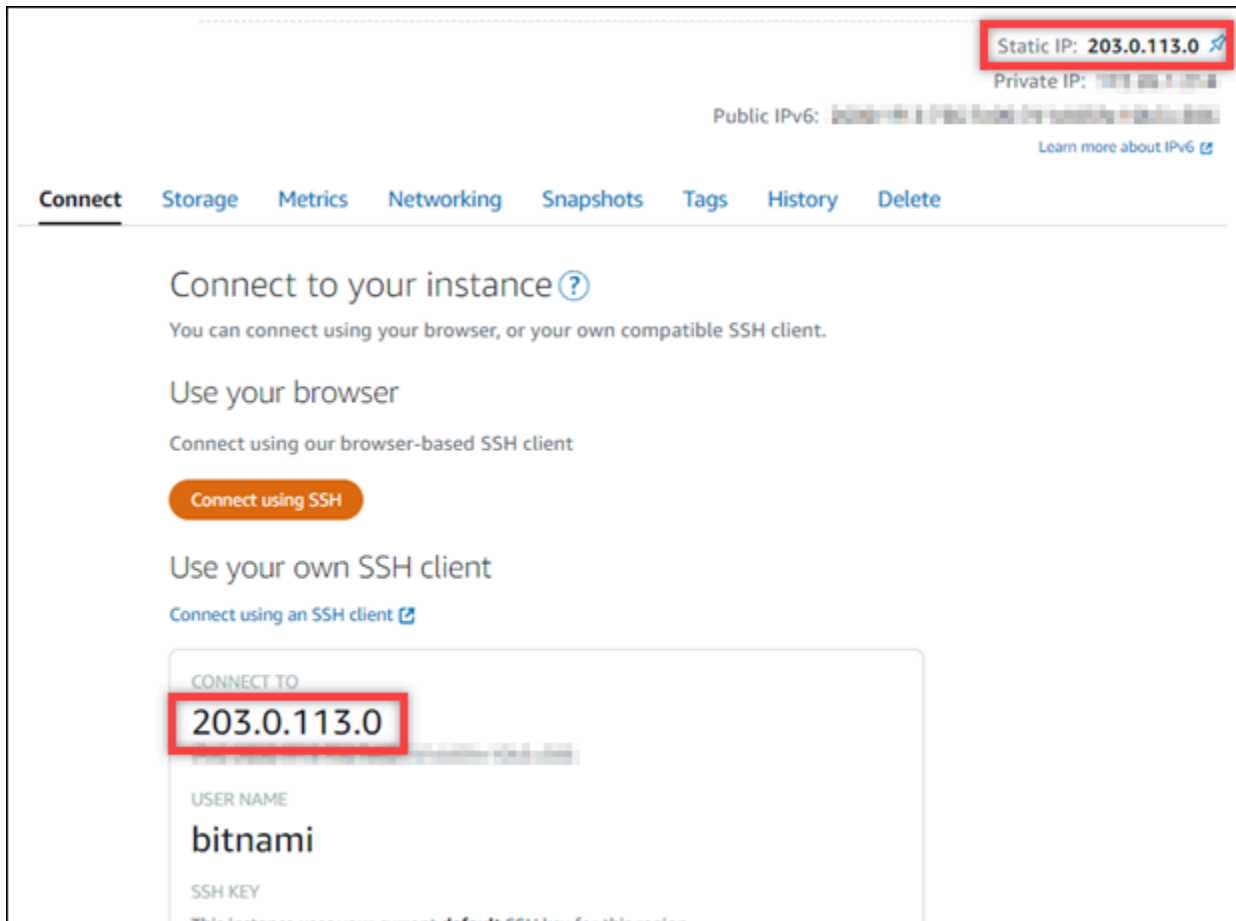
You should see a response similar to the following example. The application on your instance should now be aware of the new static IP address.

```
bitnami@ip-172-31-0-11:~$ sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
Configuring domain to 203.0.113.0
2022-06-09T16:47:06.737Z - info: Saving configuration info to disk
gitlab 16:47:06.86 INFO ==> Updating external URL in GitLab configuration
gitlab 16:47:06.88 INFO ==> Reconfiguring GitLab
gitlab 16:47:45.29 INFO ==> Starting GitLab services
Disabling automatic domain_update for IP address changes
```

Step 4: Sign in to the admin area of your Gitlab CE website

Now that you have the default user password, navigate to your GitLab CE website's home page, and sign in to the admin area. After you're signed in, you can start customizing your website and making administrative changes. For more information about what you can do in GitLab CE, see the [Step 7: Read the GitLab CE documentation and continue configuring your website](#) section later in this guide.

1. On your instance management page, under the **Connect** tab, make note of the public IP address of your instance. The public IP address is also displayed in the header section of your instance management page.

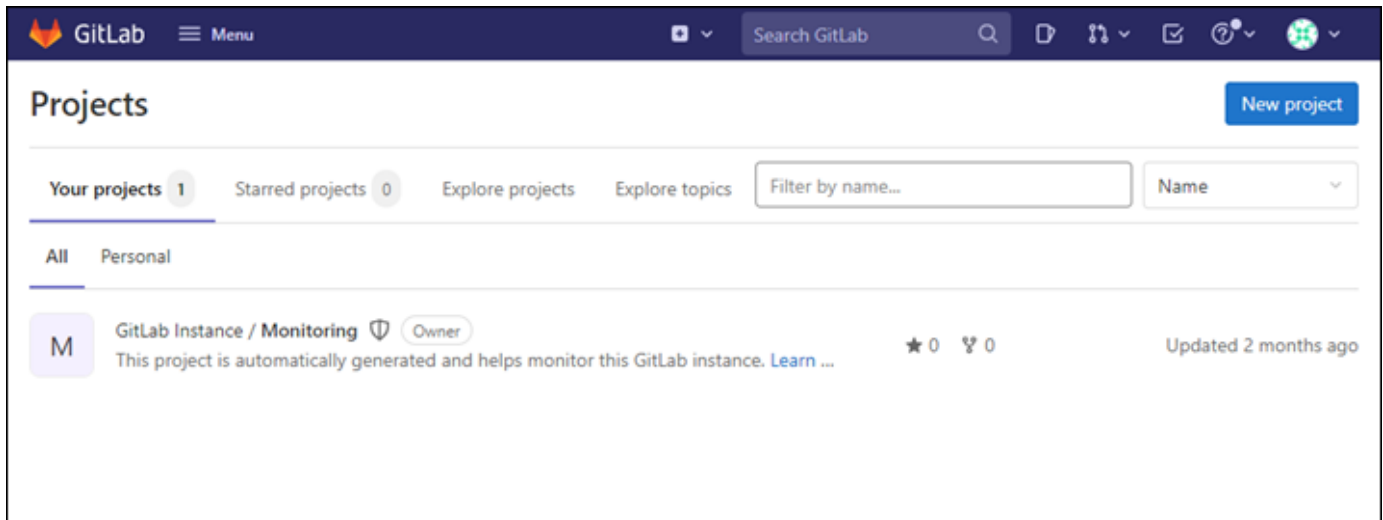


2. Browse to the public IP address of your instance, for example by going to `http://203.0.113.0`.

The home page of your Gitlab CE website should appear. You might also see a browser warning that your connection is not private, not secure, or that there's a security risk. This happens because your GitLab CE instance does not yet have an SSL/TLS certificate applied to it. In the browser window, choose **Advanced**, **Details**, or **More information** to view the options that are available. Then choose to proceed to the website even if it's not private or secure.

3. Sign in using the default user name (`root`) and the default password retrieved earlier in this guide.

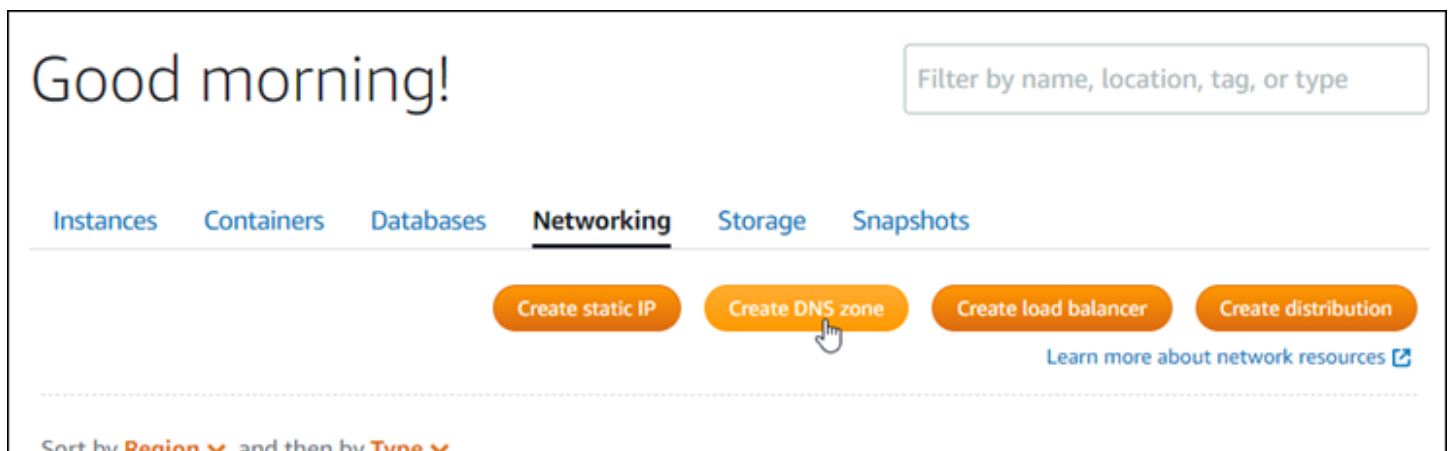
The Gitlab CE administration dashboard appears.



Step 5: Route traffic for your registered domain name to your GitLab CE website

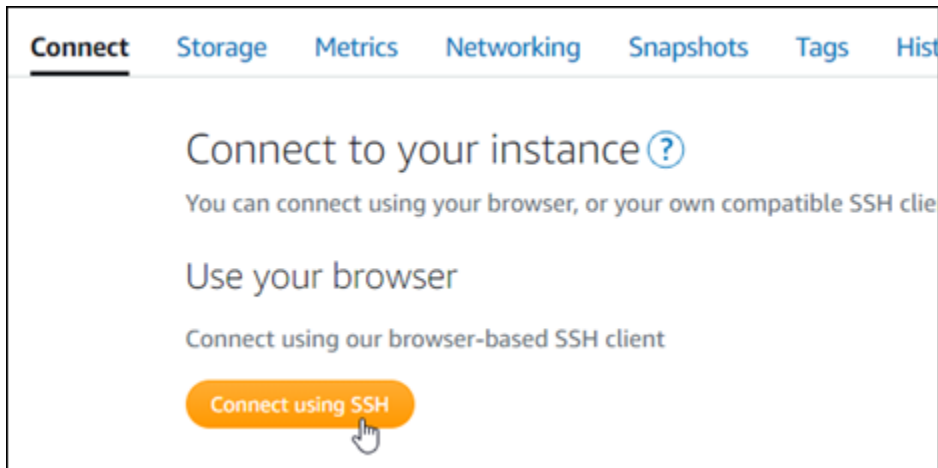
To route traffic for your registered domain name, such as `example.com`, to your GitLab CE website, you add a record to the domain name system (DNS) of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Networking** tab, choose **Create DNS zone**, then follow the instructions on the page. For more information, see [Create a DNS zone to manage your domain's DNS records](#).



After your domain name is routing traffic to your instance, you must complete the following procedure to make GitLab CE aware of the domain name.

1. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command. Replace *<DomainName>* with the domain name that is routing traffic to your instance.

```
sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain example.com
```

You should see a response similar to the following example. Your GitLab CE instance should now be aware of the domain name.

```
bitnami@ip-10.0.0.10:~$ sudo /opt/bitnami/configure_app_domain --domain example.com
Configuring domain to example.com
2022-06-09T18:44:00.235Z - info: Saving configuration info to disk
gitlab 18:44:00.36 INFO ==> Updating external URL in GitLab configuration
gitlab 18:44:00.37 INFO ==> Reconfiguring GitLab
gitlab 18:44:38.79 INFO ==> Starting GitLab services
Disabling automatic domain update for IP address changes
```

If that command fails, you might be using an older version of the GitLab CE instance. Try running the following commands instead. Replace *<DomainName>* with the domain name that is routing traffic to your instance.

```
cd /opt/bitnami/apps/gitlab
sudo ./bnconfig --machine_hostname <DomainName>
```

After running those commands, enter the following command to keep the bnconfig tool from automatically running every time the server restarts.

```
sudo mv bnconfig bnconfig.disabled
```

Next, you should generate and configure an SSL/TLS certificate to enable HTTPS connections for your GitLab CE website. For more information, continue to the next [Step 6: Configure HTTPS for your GitLab CE website](#) section of this guide.

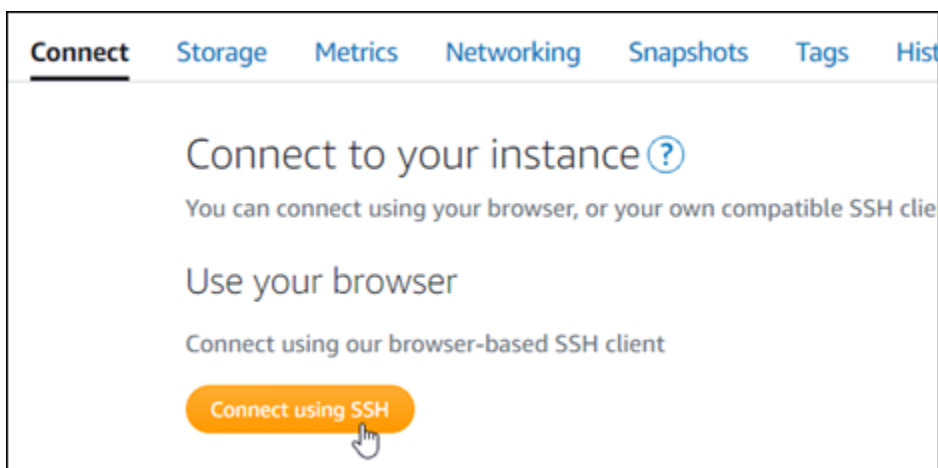
Step 6: Configure HTTPS for your GitLab CE website

Complete the following procedure to configure HTTPS on your GitLab CE website. These steps show you how to use the [Lego client](#), which is a command line tool for requesting Let's Encrypt SSL/TLS certificates.

Important

Before starting with this procedure, make sure that you configured your domain to route traffic to your GitLab CE instance. Otherwise, the SSL/TLS certificate validation process will fail. To route traffic for your registered domain name, you add a record to the DNS of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console. On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page. For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to change directory to the temporary (/tmp) directory.

```
cd /tmp
```

3. Enter the following command to download the latest version of the Lego client. This command downloads a tape archive (tar) file.

```
curl -Ls https://api.github.com/repos/xenolf/lego/releases/latest | grep  
browser_download_url | grep linux_amd64 | cut -d '"' -f 4 | wget -i -
```

4. Enter the following command to extract the files from the tar file. Replace *X.Y.Z* with the version of the Lego client that you downloaded.

```
tar xf lego_vX.Y.Z_linux_amd64.tar.gz
```

Example:

```
tar xf lego_v4.7.0_linux_amd64.tar.gz
```

5. Enter the following command to create the /opt/bitnami/letsencrypt directory where you will move the Lego client files into.

```
sudo mkdir -p /opt/bitnami/letsencrypt
```

6. Enter the following command to move the Lego client files into the directory you created.

```
sudo mv lego /opt/bitnami/letsencrypt/lego
```

7. Enter the following commands one by one to stop the application services that are running on your instance.

```
sudo service bitnami stop  
sudo service gitlab-runsvdir stop
```

8. Enter the following command to use the Lego client to request a Let's Encrypt SSL/TLS certificate.

```
sudo /opt/bitnami/letsencrypt/lego --tls --email="EmailAddress" --  
domains="RootDomain" --domains="WwwSubDomain" --path="/opt/bitnami/letsencrypt" run
```

In the command, replace the following example values with your own:

- *EmailAddress* — Your email address for registration notifications.
- *RootDomain* — The primary root domain that is routing traffic to your GitLab CE website (for example, `example.com`).
- *WwwSubDomain* — The `www` subdomain of the primary root domain that is routing traffic to your GitLab CE website (for example, `www.example.com`).

You can specify multiple domains for your certificate by specifying additional `--domains` parameters in your command. When you specify multiple domains, Leto creates a subject alternate names (SAN) certificate which results in only one certificate being valid for all domains you specified. The first domain in your list is added as the “CommonName” of the certificate and the rest are added as “DNSNames” to the SAN extension within the certificate.

Example:

```
sudo /opt/bitnami/letsencrypt/lego --tls --email="user@example.com" --  
domains="example.com" --domains="www.example.com" --path="/opt/bitnami/letsencrypt"  
run
```

9. Press **Y** and **Enter** when to accept the terms of service when prompted.

You should see a response similar to the following example.

```
2022/06/09 19:23:27 [INFO] [ example.com ] Server responded with a certificate.
```

If successful, a set of certificates are saved to the `/opt/bitnami/letsencrypt/certificates` directory. This set includes the server certificate file (for example, `example.com.crt`) and the server certificate key file for (example, `example.com.key`).

10. Enter the following commands one by one to rename the existing certificates on your instance. Later, you will replace these existing certificates with your new Let's Encrypt certificates.

```
sudo mv /etc/gitlab/ssl/server.crt /etc/gitlab/ssl/server.crt.old  
sudo mv /etc/gitlab/ssl/server.key /etc/gitlab/ssl/server.key.old  
sudo mv /etc/gitlab/ssl/server.csr /etc/gitlab/ssl/server.csr.old
```

11. Enter the following commands one by one to create symbolic links for your new Let's Encrypt certificates in the `/etc/gitlab/ssl` directory, which is the default certificates directory on your GitLab CE instance.

```
sudo ln -sf /opt/bitnami/letsencrypt/certificates/Domain.key /etc/gitlab/ssl/server.key
sudo ln -sf /opt/bitnami/letsencrypt/certificates/Domain.cert /etc/gitlab/ssl/server.crt
```

In the command, replace *Domain* with the primary root domain that you specified when requesting your Let's Encrypt certificates.

Example:

```
sudo ln -sf /opt/bitnami/letsencrypt/certificates/example.com.key /etc/gitlab/ssl/server.key
sudo ln -sf /opt/bitnami/letsencrypt/certificates/example.com.cert /etc/gitlab/ssl/server.crt
```

12. Enter the following commands one by one to change the permissions of your new Let's Encrypt certificates in the directory you moved them into.

```
sudo chown root:root /etc/gitlab/ssl/server*
sudo chmod 600 /etc/gitlab/ssl/server*
```

13. Enter the following command to restart the application services on your GitLab CE instance.

```
sudo service bitnami start
```

Next time you browse to your GitLab CE website using the domain you configured, you should see that it redirects to the HTTPS connection. Note that it can take up to an hour for the GitLab CE instance to recognize the new certificates. If your GitLab CE website refuses your connection, stop and start the instance, and try again.

Step 7: Read the GitLab CE documentation and continue configuring your website

Read the GitLab CE documentation to learn how to administer and customize your website. For more information, see the [GitLab Documentation](#).

Step 8: Create a snapshot of your instance

After you configure your GitLab CE website the way you want it, create periodic snapshots of your instance to back it up. You can create snapshots manually, or enable automatic snapshots to have Lightsail create daily snapshots for you. If something goes wrong with your instance, you can create a new replacement instance using the snapshot. For more information, see [Snapshots](#).

On the instance management page, under the **Snapshot** tab, choose **Create a snapshot** or choose to enable automatic snapshots.

The screenshot shows the 'Snapshots' tab in the Amazon Lightsail console. At the top, there are navigation tabs: Connect, Storage, Metrics, Networking, **Snapshots**, Tags, History, and Delete. The main content is divided into two sections: 'Manual snapshots' and 'Automatic snapshots'.

Manual snapshots section:

- Header: Manual snapshots (?)
- Description: You can create a snapshot to back up your instance, its system disk, and attached disks.
- Action: + Create snapshot
- Table of snapshots:

Snapshot Name	Creation Time	Snapshot ID	Actions
> <input type="checkbox"/> February 5, 2021 - 9:37 AM		"Prestashop-1612546662"	⋮
> <input type="checkbox"/> January 13, 2021 - 9:44 AM		"Prestashop-1610559880"	⋮
> <input type="checkbox"/> December 9, 2020 - 12:33 PM		"Prestashop-1607545986"	⋮
> <input type="checkbox"/> September 9, 2020 - 5:44 PM		"Prestashop-1599698658"	⋮

Showing 4 of 4 snapshots

Automatic snapshots section:

- Header: Automatic snapshots (?)
- Toggle: Automatic snapshots are enabled
- Description: Your daily snapshot time is 10:00 PM PST. We will store your seven most recent snapshots.
- Action: Change snapshot time
- Section: DAILY SNAPSHOTS
- Table of daily snapshots:

Day	Snapshot Date	Actions
> <input type="checkbox"/> Thursday	March 4, 2021	⋮
> <input type="checkbox"/> Wednesday	March 3, 2021	⋮
> <input type="checkbox"/> Tuesday	March 2, 2021	⋮

For more information, see [Creating a snapshot of your Linux or Unix instance in Amazon Lightsail](#) or [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

Quick start guide: Joomla!

Here are a few steps you should take to get started after your Joomla! instance is up and running on Amazon Lightsail:

Contents

- [Step 1: Read the Bitnami documentation](#)
- [Step 2: Get the default application password to access the Joomla! control panel](#)
- [Step 3: Attach a static IP address to your instance](#)
- [Step 4: Sign in to the control panel of your Joomla! website](#)
- [Step 5: Route traffic for your registered domain name to your Joomla! website](#)
- [Step 6: Configure HTTPS for your Joomla! website](#)
- [Step 7: Read the Joomla! documentation and continue configuring your website](#)
- [Step 8: Create a snapshot of your instance](#)

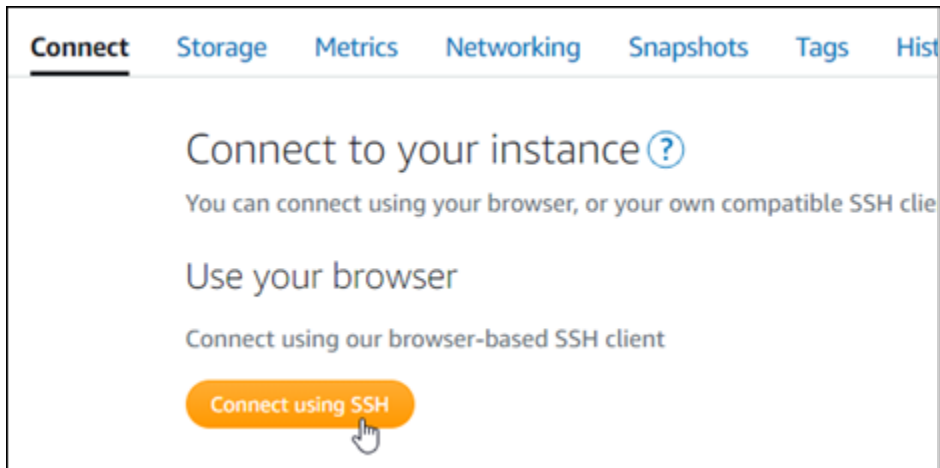
Step 1: Read the Bitnami documentation

Read the Bitnami documentation to learn how to configure your Joomla! application. For more information, see the [Joomla! Packaged By Bitnami For AWS Cloud](#).

Step 2: Get the default application password to access the Joomla! control panel

Complete the following procedure to get the default application password required to access the control panel for your Joomla! website. For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to get the application password:

```
cat $HOME/bitnami_application_password
```

You should see a response similar to the following example, which contains the default application password:

```
bitnami@ip-172-31-33-100:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-172-31-33-100:~$
```

Step 3: Attach a static IP address to your instance

The public IP address assigned to your instance when you first create it will change every time you stop and start your instance. You should create and attach a static IP address to your instance to ensure its public IP address doesn't change. Later, when you use a registered domain name, such as `example.com`, with your instance, you don't have to update your domain's DNS records every time you stop and start your instance. You can attach one static IP to an instance.

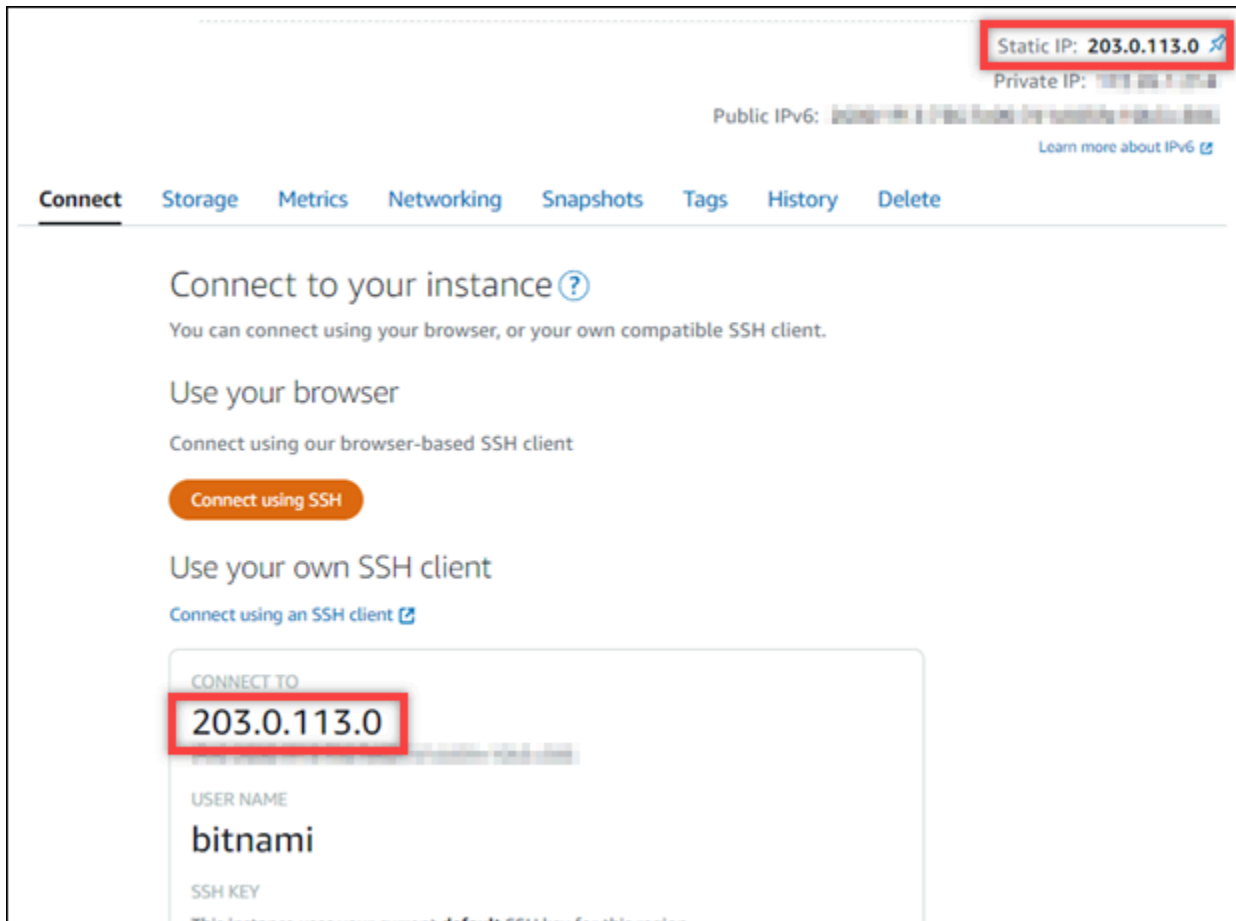
On the instance management page, under the **Networking** tab, choose **Create a static IP** or **Attach static IP** (if you previously created a static IP that you can attach to your instance), then follow the instructions on the page. For more information, see [Create a static IP and attach it to an instance](#).



Step 4: Sign in to the control panel of your Joomla! website

Now that you have the default application password, complete the following procedure to navigate to your Joomla! website's home page, and sign in to the control panel. After you're signed in, you can start customizing your website and making administrative changes. For more information about what you can do in Joomla!, see the [Step 7: Read the Joomla! documentation and continue configuring your website](#) section later in this guide.

1. On your instance management page, under the **Connect** tab, make note of the public IP address of your instance. The public IP address is also displayed in the header section of your instance management page.



2. Browse to the public IP address of your instance, for example by going to `http://203.0.113.0`.

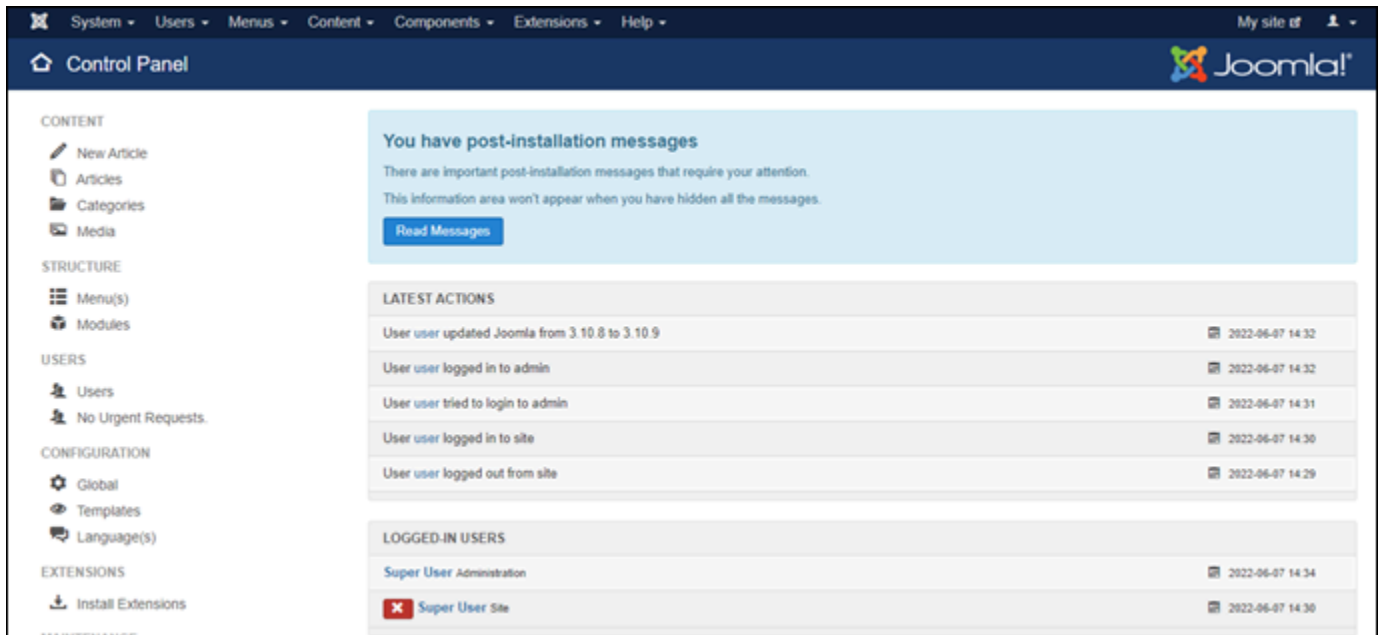
The home page of your Joomla! website should appear.

3. Choose **Manage** in the bottom right corner of your Joomla! website home page.

If the **Manage** banner is not shown, you can reach the sign in page by browsing to `http://<PublicIP>/administrator/`. Replace *<PublicIP>* with the public IP address of your instance.

4. Sign in using the default user name (`user1`) and the default password retrieved earlier in this guide.

The Joomla! administration control panel appears.



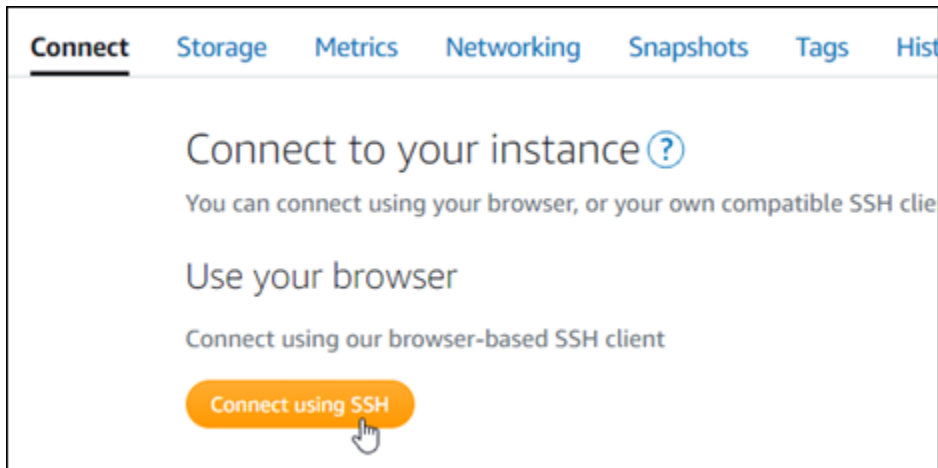
Step 5: Route traffic for your registered domain name to your Joomla! website

To route traffic for your registered domain name, such as `example.com`, to your Joomla! website, you add a record to the domain name system (DNS) of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page. For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

After your domain name is routing traffic to your instance, you must complete the following steps to make the Joomla! software aware of the domain name.

1. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. Bitnami is in the process of modifying the file structure for many of their blueprints. The file paths in this procedure may change depending on whether your Bitnami blueprint uses native Linux system packages (Approach A), or if it is a self-contained installation (Approach B). To identify your Bitnami installation type and which approach to follow, run the following command after you're connected:

```
test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach A: Using system packages." || echo "Approach B: Self-contained installation."
```

3. Complete the following steps if the result of the previous command indicated that you should use approach A. Otherwise, continue to step 4 if the result of the previous command indicated that you should use approach B.

1. Enter the following command to open the Apache virtual host configuration file using Vim and create a virtual host for your domain name.

```
sudo vim /opt/bitnami/apache2/conf/vhosts/joomla-vhost.conf
```

2. Press **I** to enter insert mode in Vim.
3. Add your domain name to the file as shown in the following example. In this example we are using the `example.com` and `www.example.com` domains.

```
<VirtualHost 127.0.0.1:80_default_:80>
  ServerName www.example.com
  ServerAlias example.com
  DocumentRoot /opt/bitnami/joomla
  <Directory "/opt/bitnami/joomla">
    Options -Indexes +FollowSymLinks -MultiViews
    AllowOverride None
    Require all granted
  </Directory>
  Include "/opt/bitnami/apache/conf/vhosts/htaccess/joomla-htaccess.conf"
</VirtualHost>
```

4. Press the **Esc** key, and enter `:wq!` to save your edit (write) and exit Vim.
5. Enter the following command to restart the Apache server.

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

4. Complete the following steps if the result of the previous command indicated that you should use approach B.

1. Enter the following command to open the Apache virtual host configuration file using Vim and create a virtual host for your domain name.

```
sudo vim /opt/bitnami/apps/joomla/conf/httpd-vhosts.conf
```

2. Press **I** to enter insert mode in Vim.
3. Add your domain name to the file as shown in the following example. In this example we are using the `example.com` and `www.example.com` domains.

```
<VirtualHost *:80>
  ServerName example.com
  ServerAlias www.example.com
  ...
```

4. Press the **Esc** key, and enter `:wq!` to save your edit (write) and exit Vim.
5. Enter the following command to confirm that the `bitnami-apps-vhosts.conf` file includes the `httpd-vhosts.conf` file for Joomla!.

```
sudo vim /opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf
```

Look for the following line in the file. Add it if it's missing.

```
Include "/opt/bitnami/apps/joomla/conf/httpd-vhosts.conf"
```

6. Enter the following command to restart the Apache server.

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

If you browse to the domain name that you configured for your instance, you should be redirected to the home page of your Joomla! website. Next, you should generate and configure an SSL/TLS certificate to enable HTTPS connections for your Joomla! website. For more information, continue to the next [Step 6: Configure HTTPS for your Joomla! website](#) section of this guide.

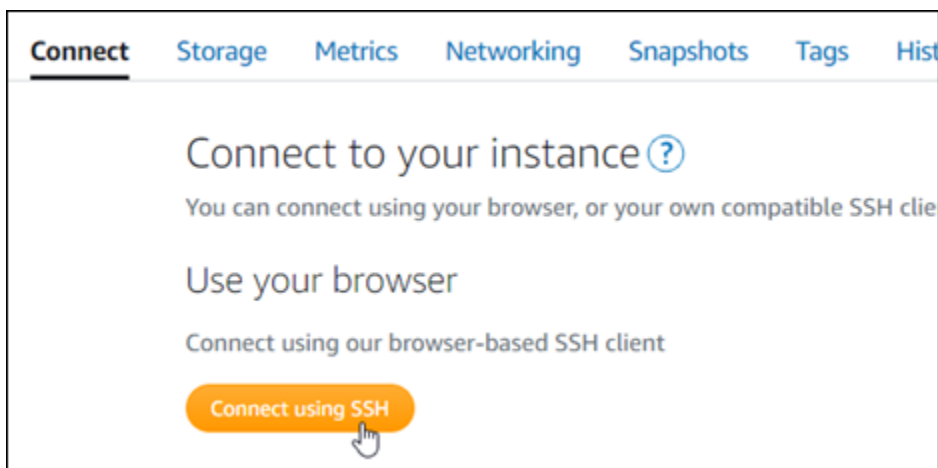
Step 6: Configure HTTPS for your Joomla! website

Complete the following procedure to configure HTTPS on your Joomla! website. These steps show you how to use the Bitnami HTTPS Configuration Tool (`bncert-tool`), which is a command line tool for requesting Let's Encrypt SSL/TLS certificates. For more information see [Learn About The Bitnami HTTPS Configuration Tool](#) in the *Bitnami documentation*.

Important

Before starting with this procedure, make sure that you configured your domain to route traffic to your Joomla! instance. Otherwise, the SSL/TLS certificate validation process will fail.

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to confirm the bncert tool is installed on your instance.

```
sudo /opt/bitnami/bncert-tool
```

You should see one of the following responses:

- If you see `command not found` in the response, then the bncert tool is not installed on your instance. Continue to the next step in this procedure to install the bncert tool on your instance.
 - If you see **Welcome to the Bitnami HTTPS configuration tool** in the response, then the bncert tool is installed on your instance. Continue to the step 8 of this procedure.
 - If the bncert tool has been installed on your instance for a while, then you might see a message indicating that an updated version of the tool is available. Choose to download it, and then enter the `sudo /opt/bitnami/bncert-tool` command to run the bncert tool again. Continue to the step 8 of this procedure.
3. Enter the following command to download the bncert run file to your instance.

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/bncert-linux-x64.run
```

4. Enter the following command to create a directory for the bncert tool run file on your instance.

```
sudo mkdir /opt/bitnami/bncert
```

5. Enter the following command to make the bncert run a file that can be executed as a program.

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. Enter the following command to create a symbolic link that runs the bncert tool when you enter the `sudo /opt/bitnami/bncert-tool` command.

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

You are now done installing the bncert tool on your instance.

7. Enter the following command to run the bncert tool.

```
sudo /opt/bitnami/bncert-tool
```

8. Enter your primary domain name and alternate domain names separated by a space as shown in the following example.

If your domain is not configured to route traffic to the public IP address of your instance, the `bncert` tool will ask you to make that configuration before continuing. Your domain must be routing traffic to the public IP address of the instance from which you are using the `bncert` tool to enable HTTPS on the instance. This confirms that you own the domain, and serves as the validation for your certificate.

```
-----
Welcome to the Bitnami HTTPS Configuration tool.
-----
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: example.com www.example.com
```

9. The `bncert` tool will ask you how you want your website's redirection to be configured. These are the options available:
 - **Enable HTTP to HTTPS redirection** - Specifies whether users who browse to the HTTP version of your website (i.e., `http://example.com`) are automatically redirected to the HTTPS version (i.e., `https://example.com`). We recommend enabling this option because it forces all visitors to use the encrypted connection. Type **Y** and press **Enter** to enable it.
 - **Enable non-www to www redirection** - Specifies whether users who browse to the apex of your domain (i.e., `https://example.com`) are automatically redirected to your domain's `www` subdomain (i.e., `https://www.example.com`). We recommend enabling this option. However, you may want to disable it and enable the alternate option (enable `www` to non-`www` redirection) if you have specified the apex of your domain as your preferred website address in search engine tools like Google's webmaster tools, or if your apex points directly to your IP and your `www` subdomain references your apex via a CNAME record. Type **Y** and press **Enter** to enable it.
 - **Enable www to non-www redirection** - Specifies whether users who browse to your domain's `www` subdomain (i.e., `https://www.example.com`) are automatically redirected to the apex of your domain (i.e., `https://example.com`). We recommend disabling this, if you enabled non-`www` redirection to `www`. Type **N** and press **Enter** to disable it.

Your selections should look like the following example.

```
Enable/disable redirections

Please select the redirections you wish to enable or disable on your Bitnami
installation.

Enable HTTP to HTTPS redirection [Y/n]: Y

Enable non-www to www redirection [Y/n]: Y

Enable www to non-www redirection [y/N]: N
```

10. The changes that are going to be made are listed. Type Y and press **Enter** to confirm and continue.

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
www.example.com)
7. Start web server once all changes have been performed

Do you agree to these changes? [Y/n]: Y
```

11. Enter your email address to associate with your Let's Encrypt certificate and press **Enter**.

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []: 
```

12. Review the Let's Encrypt Subscriber Agreement. Type **Y** and press **Enter** to accept the agreement and continue.

```
The Let's Encrypt Subscriber Agreement can be found at:  
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf  
Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]: █
```

The actions are performed to enable HTTPS on your instance, including requesting the certificate and configuring the redirections you specified.

```
Performing changes to your installation  
  
The Bitnami HTTPS Configuration Tool will perform any necessary actions to your  
Bitnami installation. This may take some time, please be patient.  
  
█
```

Your certificate is successfully issued and validated, and the redirections are successfully configured on your instance if you see a message similar to the following example.

```
Success  
  
The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.  
The configuration report is shown below.  
  
Backup files:  
* /opt/bitnami/apache2/conf/httpd.conf.back.202005290035  
* /opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf.back.202005290035  
* /opt/bitnami/apache2/conf/bitnami/bitnami.conf.back.202005290035  
  
Find more details in the log file:  
/tmp/bncert-202005290035.log  
  
If you find any issues, please check Bitnami Support forums at:  
https://community.bitnami.com  
  
Press [Enter] to continue:█
```

The `bncert` tool will perform an automatic renewal of your certificate every 80 days before it expires. Repeat the above steps if you wish to use additional domains and subdomains with your instance, and you want to enable HTTPS for those domains.

You are now done enabling HTTPS on your Joomla! instance. Next time you browse to your Joomla! website using the domain you configured, you should see that it redirects to the HTTPS connection.

Step 7: Read the Joomla! documentation and continue configuring your website

Read the Joomla! documentation to learn how to administer and customize your website. For more information, see the [Joomla! Documentation](#).

Step 8: Create a snapshot of your instance

After you configure your Joomla! website the way you want it, create periodic snapshots of your instance to back it up. You can create snapshots manually, or enable automatic snapshots to have Lightsail create daily snapshots for you. If something goes wrong with your instance, you can create a new replacement instance using the snapshot. For more information, see [Snapshots](#).

On the instance management page, under the **Snapshot** tab, choose **Create a snapshot** or choose to enable automatic snapshots.

Connect
Storage
Metrics
Networking
Snapshots
Tags
History
Delete

Manual snapshots ?

You can create a snapshot to back up your instance, its system disk, and attached disks.

[+ Create snapshot](#)

>		February 5, 2021 - 9:37 AM	"Prestashop-1612546662"	⋮
>		January 13, 2021 - 9:44 AM	"Prestashop-1610559880"	⋮
>		December 9, 2020 - 12:33 PM	"Prestashop-1607545986"	⋮
>		September 9, 2020 - 5:44 PM	"Prestashop-1599698658"	⋮

Showing 4 of 4 snapshots

Automatic snapshots ?

Automatic snapshots are enabled

Your daily snapshot time is 10:00 PM PST.
We will store your seven most recent snapshots.

[Change snapshot time](#)

DAILY SNAPSHOTS

>		Thursday	March 4, 2021	⋮
>		Wednesday	March 3, 2021	⋮
>		Tuesday	March 2, 2021	⋮

For more information, see [Creating a snapshot of your Linux or Unix instance in Amazon Lightsail](#) or [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

Quick start guide: LAMP

Here are a few steps you should take to get started after your LAMP instance is up and running on Amazon Lightsail:

Step 1: Get the default application password for your LAMP instance

You need the default application password to access pre-installed applications or services on your instance.

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.
2. After you're connected, enter the following command to get the application password:

```
cat bitnami_application_password
```

Note

If you're in a directory other than the user home directory, then enter `cat $HOME/bitnami_application_password`.

You should see a response similar to this, which contains the default application password:

```
bitnami@ip-172-31-33-100:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-172-31-33-100:~$
```

For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

Step 2: Attach a static IP address to your LAMP instance

The default dynamic public IP address attached to your instance changes every time you stop and start the instance. Create a static IP address, and attach it to your instance, to keep the public IP address from changing. Later, when you use your domain name with your instance, you don't have to update your domain's DNS records each time you stop and start the instance. You can attach one static IP to an instance.

On your instance management page, under the **Networking** tab, choose **Create static IP**, then follow the instructions on the page.

For more information, see [Create a static IP and attach it to an instance](#).

Step 3: Visit your LAMP instance welcome page

Navigate to the public IP address of your instance to access the application installed on it, access phpMyAdmin, or access the Bitnami documentation.

1. On your instance management page, under the **Connect** tab, make note of the public IP.

2. Browse to the public IP address, for example by going to `http://192.0.2.3`.

For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

Step 4: Map your domain name to your LAMP instance

To map your domain name, such as `example.com`, to your instance, you add a record to the domain name system (DNS) of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page.

For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

Step 5: Read the Bitnami documentation

Read the Bitnami documentation to learn how to deploy your application, enable HTTPs support with SSL certificates, upload files to the server with SFTP, and more.

For more information, see the [Bitnami LAMP for AWS Cloud](#).

Step 6: Create a snapshot of your LAMP instance

A snapshot is a copy of the system disk and original configuration of an instance. The snapshot includes such information as memory, CPU, disk size, and data transfer rate. You can use a snapshot as a baseline for new instances, or as a data backup.

Under the **Snapshot** tab of your instance's management page, enter a name for the snapshot, then choose **Create snapshot**.

For more information, see [Create a snapshot of your Linux or Unix instance](#).

Quick start guide: Magento

Here are a few steps you should complete to get started after your Magento instance is up and running on Amazon Lightsail.

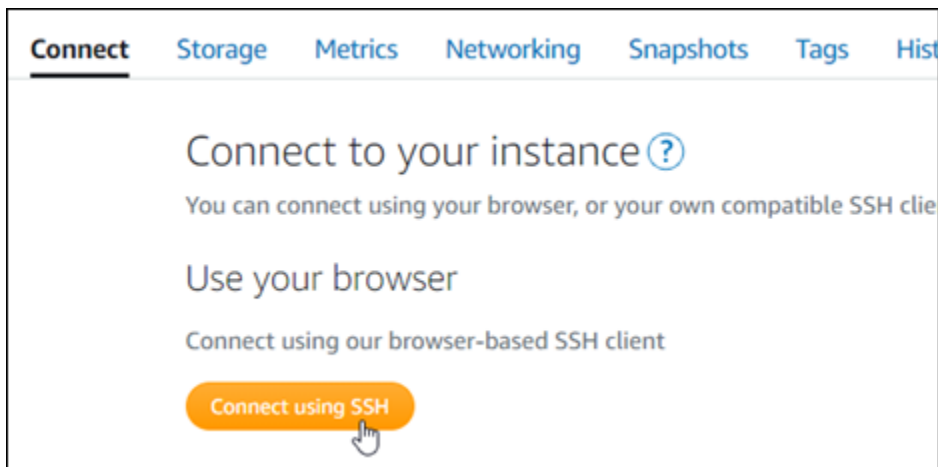
Contents

- [Step 1: Get the default application password for your Magento website](#)
- [Step 2: Attach a static IP address to your Magento instance](#)
- [Step 3: Sign in to the administration dashboard of your Magento website](#)
- [Step 4: Route traffic for your registered domain name to your Magento website](#)
- [Step 5: Configure HTTPS for your Magento website](#)
- [Step 6: Configure SMTP for email notifications](#)
- [Step 7: Read the Bitnami and Magento documentation](#)
- [Step 8: Create a snapshot of your Magento instance](#)

Step 1: Get the default application password for your Magento website

Complete the following steps to get the default application password for your Magento website. For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

1. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.

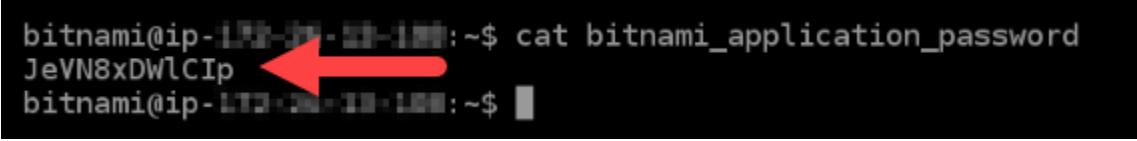


2. After you're connected, enter the following command to get the default application password:

```
cat $HOME/bitnami_application_password
```

You should see a response similar to the following example, which contains the default application password. Store this password in a safe place. You will use it in the next section of this tutorial to sign in to the administration dashboard of your Magento website.

```
bitnami@ip-192-0-2-0:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-192-0-2-0:~$
```



Step 2: Attach a static IP address to your Magento instance

The public IP address assigned to your instance when you first create it will change every time you stop and start your instance. You should create and attach a static IP address to your instance to ensure its public IP address doesn't change. Later, when you use a registered domain name, such as `example.com`, with your instance, you don't have to update your domain's DNS records every time you stop and start your instance. You can attach one static IP to an instance.

On the instance management page, under the **Networking** tab, choose **Create a static IP** or **Attach static IP** (if you previously created a static IP that you can attach to your instance), then follow the instructions on the page. For more information, see [Create a static IP and attach it to an instance](#).

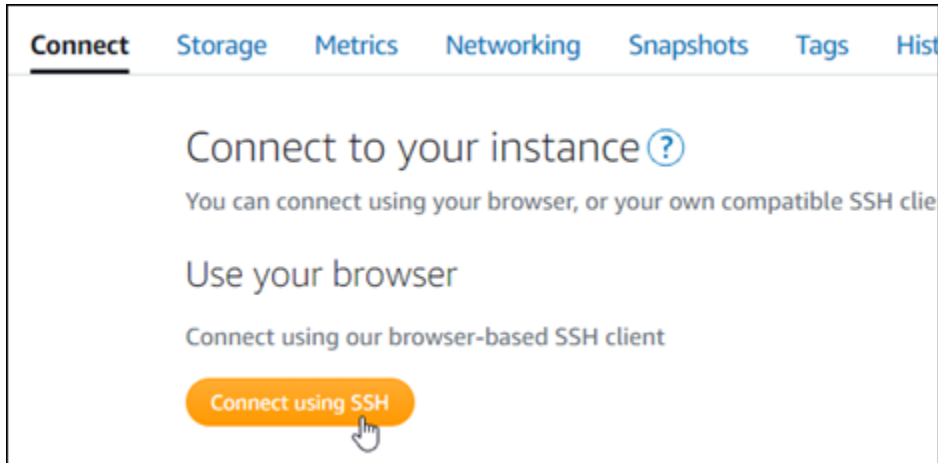


After the new static IP address is attached to your instance, you must complete the following steps to make the Magento software aware of the new static IP address.

1. Make a note of the static IP address of your instance. It's listed in the header section of your instance management page.



2. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



3. After you're connected, enter the following command. Be sure to replace *<StaticIP>* with the new static IP address of your instance.

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

You should see a response similar to the following example. The Magento software should now be aware of the new static IP address.

```
bitnami@ip-173-36-0-100:~$ sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
Configuring domain to 203.0.113.0
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO ==> Updating hostname in database
prestashop 15:49:22.46 INFO ==> Purging cache
Disabling automatic domain update for IP address changes
```

Note

Magento does not currently support IPv6 addresses. You can enable IPv6 for the instance, but the Magento software will not respond to requests over the IPv6 network.

Step 3: Sign in to the administration dashboard of your Magento website

Complete the following step to access your Magento website and sign in to its administration dashboard. To sign in, you will use the default user name (user1) and the default application password that you got earlier in this guide.

1. In the Lightsail console, make note of the public or static IP address that is listed in the header area of the instance management page.



2. Browse to the following address to access the sign in page for the administration dashboard of your Magento website. Be sure to replace *<InstanceIpAddress>* with the public or static IP address of your instance.

```
http://<InstanceIpAddress>/admin
```

Example:

```
http://203.0.113.0/admin
```

Note

You might need to reboot the instance if you can't access the sign in page for the Magento administration dashboard.

3. Enter the default user name (user), the default application password you got earlier in this guide, and choose **Sign in**.

A screenshot of the Magento login page. At the top left is the Magento logo, which consists of an orange hexagon with a white 'M' inside, followed by the word 'Magento' in a grey sans-serif font. Below the logo is the text 'Welcome, please sign in' in orange. Underneath is a form with two fields: 'Username *' with a white input box containing the text 'user', and 'Password *' with a white input box containing a series of asterisks. Below the password field is a blue link that says 'Forgot your password?'. At the bottom of the form is a blue rectangular button with the text 'Sign in' in white. A white mouse cursor is pointing at the bottom-left corner of the 'Sign in' button.

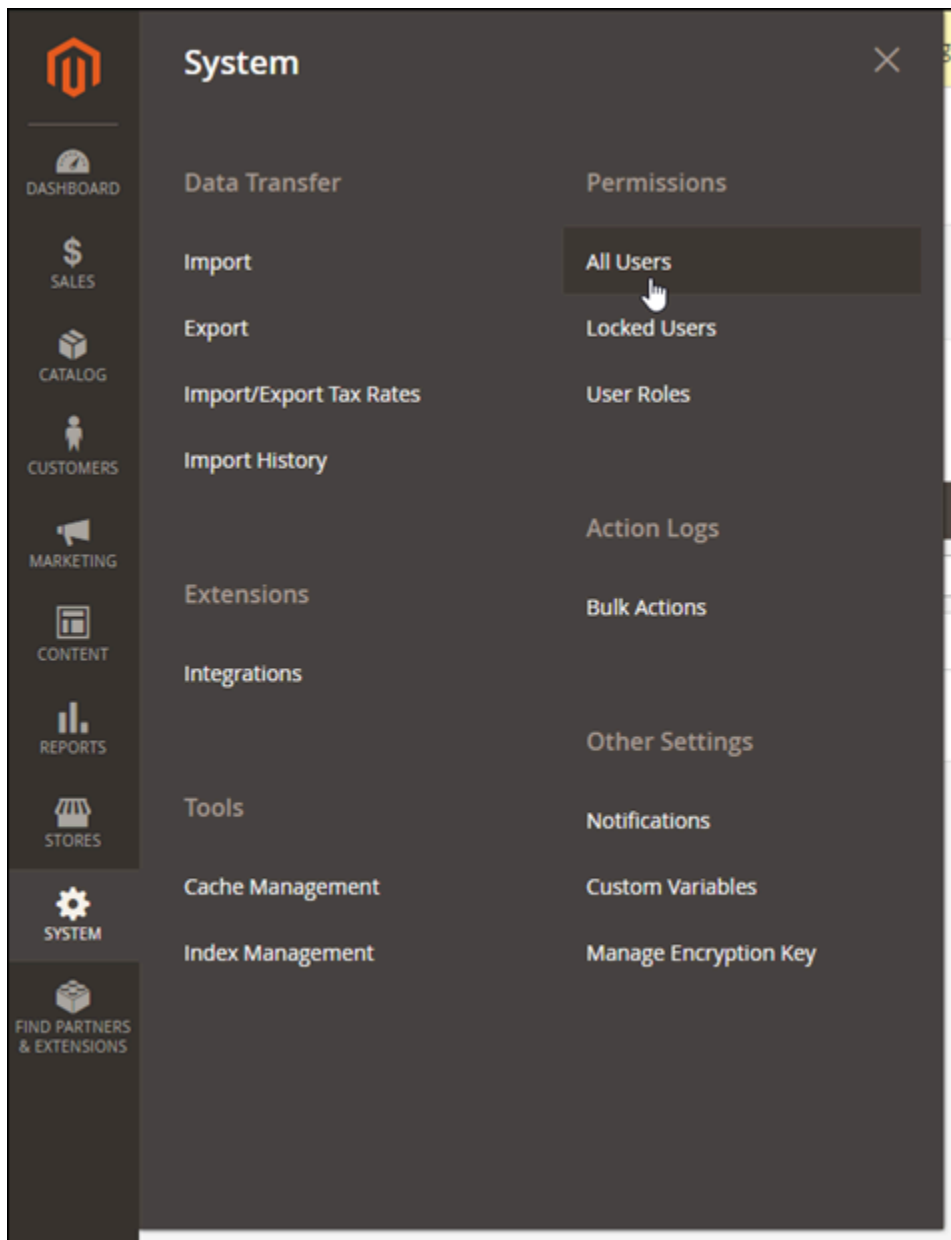
The Magento administration dashboard appears.

The screenshot shows the Magento administration dashboard. At the top, a yellow system message banner states: "One or more of the Cache Types are invalidated: Configuration. Please go to [Cache Management](#) and refresh cache". To the right of this message is a "System Messages: 1" dropdown. Below the banner is the "Dashboard" title, a search icon, a notification bell with a red "1", and a user profile icon labeled "user". A "Scope: All Store Views" dropdown with a help icon and a "Reload Data" button are also present. Another yellow banner below reads: "All other open sessions for this account were terminated." The "Advanced Reporting" section includes a description: "Gain new insights and take command of your business' performance, using our dynamic product, order, and customer reports tailored to your customer data." and a "Go to Advanced Reporting" button. The "Lifetime Sales" section shows a chart that is disabled, with a link to "enable the chart". Below this is a table with the following data:

	Revenue	Tax	Shipping	Quantity
Lifetime Sales	\$0.00	\$0.00	\$0.00	0
Average Order	\$0.00	\$0.00	\$0.00	0

The left navigation pane contains icons for: DASHBOARD, SALES, CATALOG, CUSTOMERS, MARKETING, CONTENT, REPORTS, STORES, and SYSTEM.

To change the default user name or password that you use to sign in to the administration dashboard of your Magento website, choose **System** in the navigation pane, and then choose **All Users**. For more information, see [Adding users](#) in the *Magento documentation*.



For more information about the administration dashboard, see [Magento 2.4 User Guide](#) .

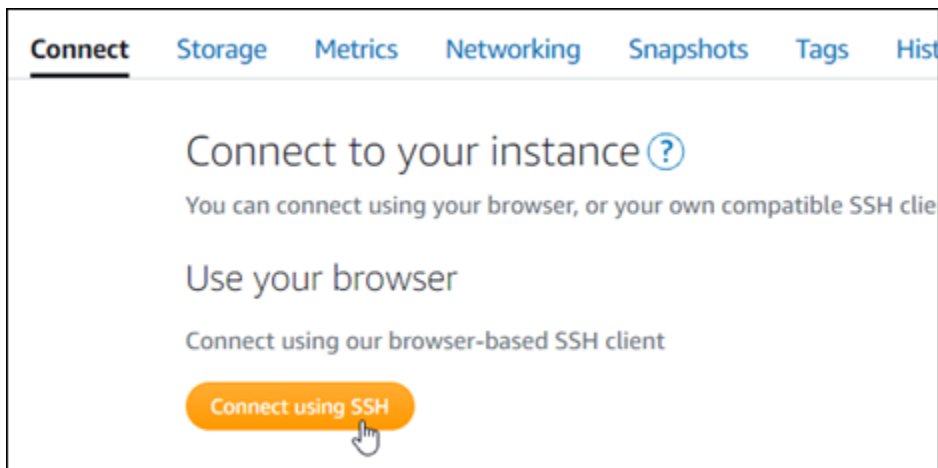
Step 4: Route traffic for your registered domain name to your Magento website

To route traffic for your registered domain name, such as `example.com`, to your Magento website, you add a record to the domain name system (DNS) of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page. For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

After your domain name is routing traffic to your instance, you must complete the following steps to make the Magento software aware of the domain name.

1. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command. Be sure to replace *<DomainName>* with the domain name that is routing traffic to your instance.

```
sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain www.example.com
```

You should see a response similar to the following example. The Magento software should now be aware of the domain name.

```
bitnami@ip-173-20-0-197:~$ sudo /opt/bitnami/configure_app_domain --domain www.example.com
Configuring domain to www.example.com
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO ==> Updating hostname in database
prestashop 15:49:22.46 INFO ==> Purging cache
Disabling automatic domain update for IP address changes
```

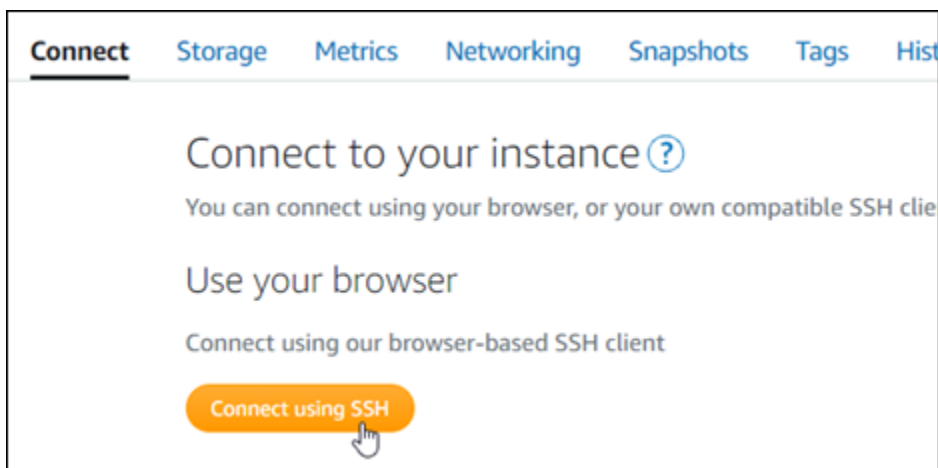
Step 5: Configure HTTPS for your Magento website

Complete the following steps to configure HTTPS on your Magento website. These steps show you how to use the Bitnami HTTPS configuration tool (bncert), which is a command line tool for requesting SSL/TLS certificates, setting up redirections (e.g. HTTP to HTTPS), and renewing certificates.

⚠ Important

The bncert tool will issue certificates only for domains that are currently routing traffic to the public IP address of your Magento instance. Before starting with these steps, make sure that you add DNS records to the DNS of all domains that you want to use with your Magento website.

1. On the instance management page, under the Connect tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to start the bncert-tool.

```
sudo /opt/bitnami/bncert-tool
```

You should see a response similar to the following example:

```
bitnami@ip-172-31-7-149:~$ sudo /opt/bitnami/bncert-tool
Warning: Custom redirections are not supported in the Bitnami Magento Stack.
This tool will not be able to enable/disable redirections.
Press [Enter] to continue:
```

3. Enter your primary domain name and alternate domain names separated by a space as shown in the following example.

```
-----  
Welcome to the Bitnami HTTPS Configuration tool.  
-----  
Domains  
Please provide a valid space-separated list of domains for which you wish to  
configure your web server.  
Domain list []: example.com www.example.com
```

4. The changes that are going to be made are listed. Type Y and press **Enter** to confirm and continue.

```
-----  
Changes to perform  
The following changes will be performed to your Bitnami installation:  
1. Stop web server  
2. Configure web server to use a free Let's Encrypt certificate for the domains:  
   example.com www.example.com  
3. Configure a cron job to automatically renew the certificate each month  
4. Configure web server name to: example.com  
5. Start web server once all changes have been performed  
  
Do you agree to these changes? [Y/n]: Y
```

5. Enter your email address to associate with your Let's Encrypt certificate and press **Enter**.

```
Create a free HTTPS certificate with Let's Encrypt  
Please provide a valid e-mail address for which to associate your Let's Encrypt  
certificate.  
Domain list: example.com www.example.com  
Server name: example.com  
E-mail address []:
```

6. Review the Let's Encrypt Subscriber Agreement. Type Y and press **Enter** to accept the agreement and continue.

```
The Let's Encrypt Subscriber Agreement can be found at:  
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf  
Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]:
```


The actions are performed to enable HTTPS on your instance, including requesting the certificate and configuring the redirections you specified.

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.
```

Your certificate is successfully issued and validated, and the redirections are successfully configured on your instance if you see a message similar to the following example.

```
Success

The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.

The configuration report is shown below.

Backup files:
* /opt/bitnami/apache/conf/httpd.conf.back.202104052147
* /opt/bitnami/apache/conf/bitnami/bitnami.conf.back.202104052147
* /opt/bitnami/apache/conf/bitnami/bitnami-ssl.conf.back.202104052147
* /opt/bitnami/apache/conf/vhosts/magento-https-vhost.conf.back.202104052147
* /opt/bitnami/apache/conf/vhosts/magento-vhost.conf.back.202104052147

Find more details in the log file:

/tmp/bncert-202104052147.log

If you find any issues, please check Bitnami Support forums at:

https://community.bitnami.com

Press [Enter] to continue:

bitnami@ip-172.31.3-143:~$
```

The bncert tool will perform an automatic renewal of your certificate every 80 days before it expires. Continue to the next set of steps to finish enabling HTTPS on your Magento website.

7. Browse to the following address to access the sign in page for the administration dashboard of your Magento website. Be sure to replace *<DomainName>* with the registered domain name that is routing traffic to your instance.

```
http://<DomainName>/admin
```

Example:

`http://www.example.com/admin`

8. Enter the default user name (user), the default application password you got earlier in this guide, and choose **Sign in**.



The Magento administration dashboard appears.

The screenshot shows the Amazon Lightsail dashboard interface. On the left is a vertical navigation sidebar with icons for Dashboard, Sales, Catalog, Customers, Marketing, Content, Reports, Stores, and System. The main content area features a top message bar with a warning about cache types. Below this is the 'Dashboard' title and a search bar. A 'Scope: All Store Views' filter is present with a 'Reload Data' button. Another warning message states that all other open sessions for the account were terminated. The 'Advanced Reporting' section includes a description and a 'Go to Advanced Reporting' button. At the bottom, a summary table displays 'Lifetime Sales' and 'Average Order' metrics.

One or more of the Cache Types are invalidated: Configuration. Please go to [Cache Management](#) and refresh cache types. System Messages: 1

Dashboard

Scope: All Store Views [?](#) [Reload Data](#)

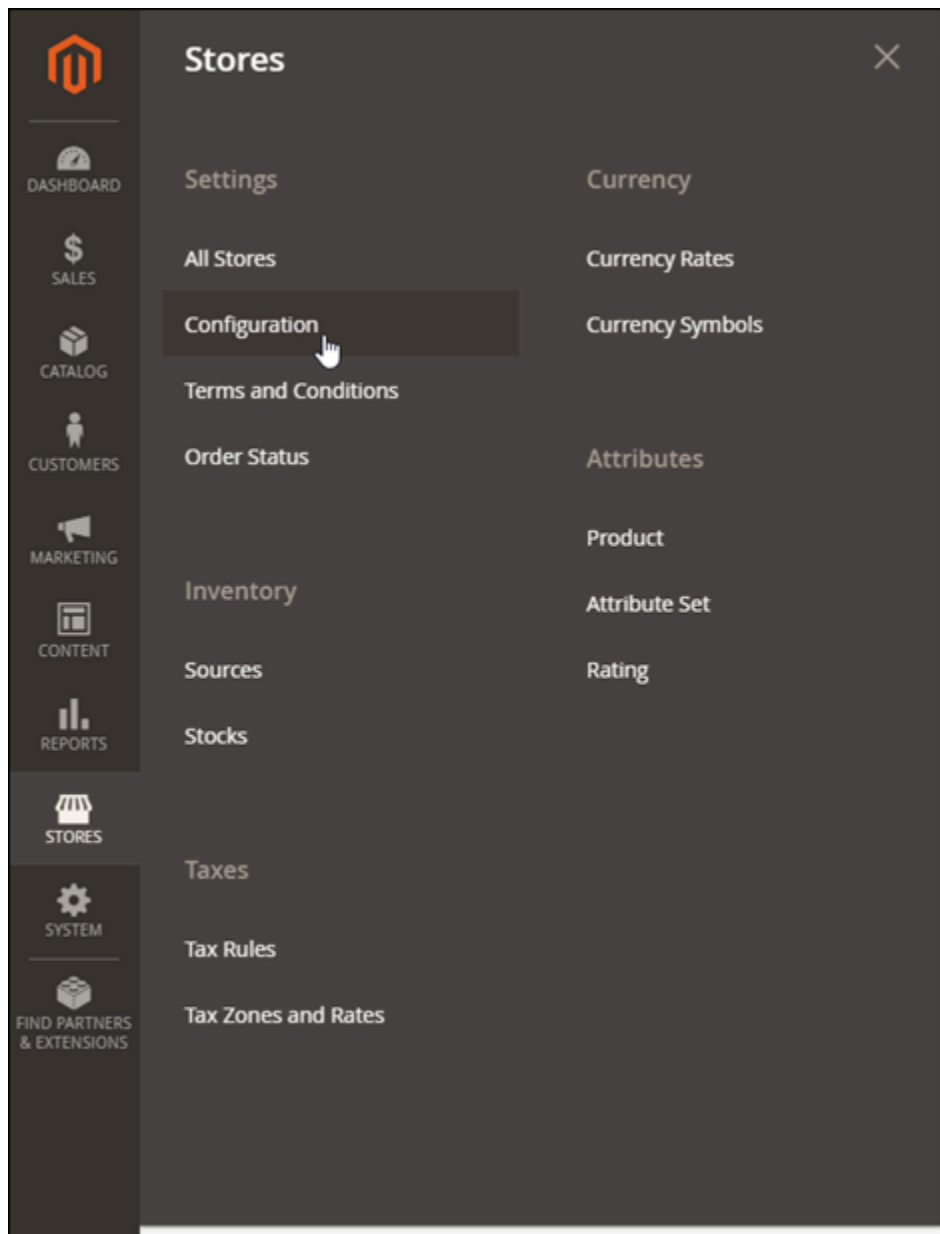
All other open sessions for this account were terminated.

Advanced Reporting

Gain new insights and take command of your business' performance, using our dynamic product, order, and customer reports tailored to your customer data. [Go to Advanced Reporting](#)

Lifetime Sales	Chart is disabled. To enable the chart, click here .			
\$0.00	Revenue	Tax	Shipping	Quantity
Average Order	\$0.00	\$0.00	\$0.00	0
\$0.00				

9. Choose **Stores** in the navigation pane, and then choose **Configuration**.



10. Choose **Web**, and then expand the **Base URLs** node.
11. In the **Base URL** text box, enter the full URL of your website, for example `https://www.example.com/`.

Base URLs

Any of the fields allow fully qualified URLs that end with '/' (slash) e.g. `http://example.com/magento/`

Base URL
[store view]
Specify URL or `{{base_url}}` placeholder.

Base Link URL
[store view] Use system value
May start with `{{unsecure_base_url}}` placeholder.

Base URL for Static View Files
[store view]
May be empty or start with `{{unsecure_base_url}}` placeholder.

Base URL for User Media Files
[store view]
May be empty or start with `{{unsecure_base_url}}` placeholder.

12. Expand the Base URLs (Secure) node.

13. In the **Secure Base URL** text box, enter the full URL of your website, for example `https://www.example.com/`.

Base URLs (Secure)

Any of the fields allow fully qualified URLs that end with '/' (slash) e.g. `https://example.com/magento/`

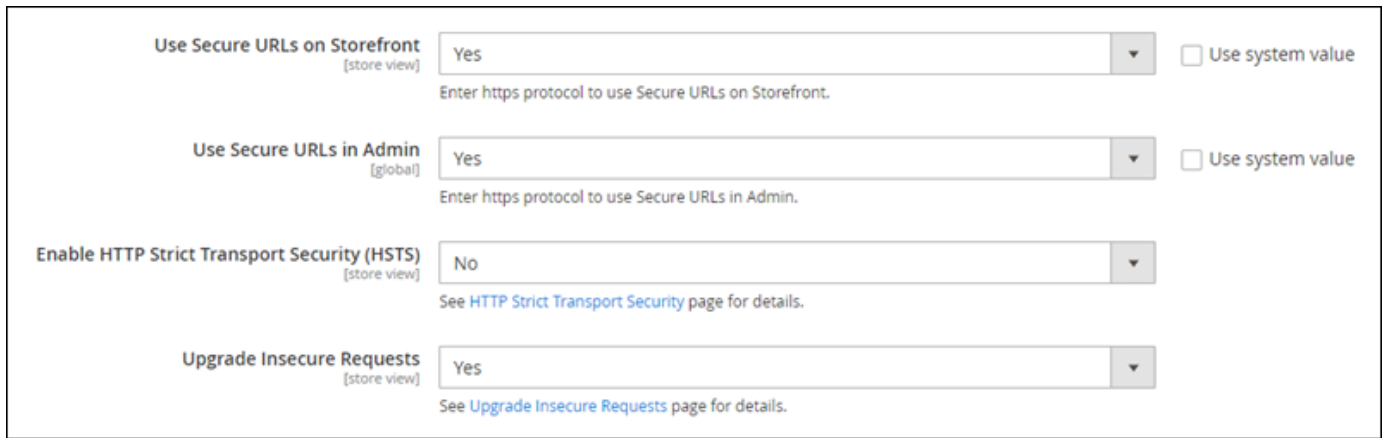
Secure Base URL
[store view]
Specify URL or `{{base_url}}`, or `{{unsecure_base_url}}` placeholder.

Secure Base Link URL
[store view] Use system value
May start with `{{secure_base_url}}` or `{{unsecure_base_url}}` placeholder.

Secure Base URL for Static View Files
[store view]
May be empty or start with `{{secure_base_url}}`, or `{{unsecure_base_url}}` placeholder.

Secure Base URL for User Media Files
[store view]
May be empty or start with `{{secure_base_url}}`, or `{{unsecure_base_url}}` placeholder.

14. Choose **Yes** for the **Use Secure URLs on Storefront**, **Use Secure URLs in Admin**, and **Upgrade Insecure Requests** options.



The screenshot shows a configuration interface with four rows of settings:

- Use Secure URLs on Storefront** [store view]: A dropdown menu is set to "Yes". Below it, a text input field contains "https" and the instruction "Enter https protocol to use Secure URLs on Storefront." To the right is a checkbox labeled "Use system value".
- Use Secure URLs in Admin** [global]: A dropdown menu is set to "Yes". Below it, a text input field contains "https" and the instruction "Enter https protocol to use Secure URLs in Admin." To the right is a checkbox labeled "Use system value".
- Enable HTTP Strict Transport Security (HSTS)** [store view]: A dropdown menu is set to "No". Below it, the instruction "See [HTTP Strict Transport Security](#) page for details." is displayed.
- Upgrade Insecure Requests** [store view]: A dropdown menu is set to "Yes". Below it, the instruction "See [Upgrade Insecure Requests](#) page for details." is displayed.

15. Choose Save Config at the top of the page.

HTTPS is now configured for your Magento website. When customers browse to the HTTP version (e.g., `http://www.example.com`) of your Magento website, they will be automatically redirected to the HTTPS version (e.g., `https://www.example.com`).

Step 6: Configure SMTP for email notifications

Configure the SMTP settings of your Magento website to enable email notifications for it. For more information, see [Install the Magento Magepal SMTP extension](#) in the *Bitnami documentation*.

Important

If you configure SMTP to use ports 25, 465, or 587, then you must open those ports in the firewall of your instance in the Lightsail console. For more information, see [Adding and editing instance firewall rules in Amazon Lightsail](#).

If you configure your Gmail account to send email on your Magento website, then you must use an app password instead of using the standard password that you use to sign in to Gmail. For more information, see [Sign in with App Passwords](#).

Step 7: Read the Bitnami and Magento documentation

Read the Bitnami documentation to learn how to perform administrative tasks on your Magento instance and website, such as install plugins and customize the theme. For more information, see [Bitnami Magento Stack for AWS Cloud](#) in the *Bitnami documentation*.

You should also read the Magento documentation to learn how to administer your Magento website. For more information, see the [Magento 2.4 User Guide](#).

Step 8: Create a snapshot of your Magento instance

After you configure your Magento website the way you want it, create periodic snapshots of your instance to back it up. You can create snapshots manually, or enable automatic snapshots to have Lightsail create daily snapshots for you. If something goes wrong with your instance, you can create a new replacement instance using the snapshot. For more information, see [Snapshots](#).

On the instance management page, under the **Snapshots** tab, choose **Create a snapshot** or choose to enable automatic snapshots.

The screenshot displays the 'Snapshots' tab in the Amazon Lightsail console. At the top, there are navigation tabs: Connect, Storage, Metrics, Networking, **Snapshots**, Tags, History, and Delete. The main content is divided into two sections: 'Manual snapshots' and 'Automatic snapshots'.

Manual snapshots section:

- Header: Manual snapshots (?)
- Description: You can create a snapshot to back up your instance, its system disk, and attached disks.
- Action: + Create snapshot
- Table of snapshots:

Snapshot Name	Creation Time	Snapshot ID	Actions
> <input type="checkbox"/> February 5, 2021 - 9:37 AM		"Prestashop-1612546662"	⋮
> <input type="checkbox"/> January 13, 2021 - 9:44 AM		"Prestashop-1610559880"	⋮
> <input type="checkbox"/> December 9, 2020 - 12:33 PM		"Prestashop-1607545986"	⋮
> <input type="checkbox"/> September 9, 2020 - 5:44 PM		"Prestashop-1599698658"	⋮

Showing 4 of 4 snapshots

Automatic snapshots section:

- Header: Automatic snapshots (?)
- Status: Automatic snapshots are enabled
- Description: Your daily snapshot time is 10:00 PM PST. We will store your seven most recent snapshots.
- Action: Change snapshot time
- Section: DAILY SNAPSHOTS
- Table of daily snapshots:

Day	Snapshot Date	Actions
> <input type="checkbox"/> Thursday	March 4, 2021	⋮
> <input type="checkbox"/> Wednesday	March 3, 2021	⋮
> <input type="checkbox"/> Tuesday	March 2, 2021	⋮

For more information, see [Creating a snapshot of your Linux or Unix instance in Amazon Lightsail](#) or [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

Quick start guide: Nginx

Here are a few steps you should take to get started after your Nginx instance is up and running on Amazon Lightsail:

Step 1: Get the default application password for your Nginx instance

You need the default application password to access pre-installed applications or services on your instance.

Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.
2. After you're connected, enter the following command to get the default application password:


```
cat bitnami_application_password
```

Note

If you're in a directory other than the user home directory, then enter `cat $HOME/bitnami_application_password`.

You should see a response similar to this, which contains the default application password:

```
bitnami@ip-172-31-33-100:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-172-31-33-100:~$
```



For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

Step 2: Attach a static IP address to your Nginx instance

The default dynamic public IP address attached to your instance changes every time you stop and start the instance. Create a static IP address, and attach it to your instance, to keep the public IP address from changing. Later, when you use your domain name with your instance, you don't have to update your domain's DNS records each time you stop and start the instance. You can attach one static IP to an instance.

On your instance management page, under the **Domains & DNS** tab, choose **Create static IP**, then follow the instructions on the page.

For more information, see [Create a static IP and attach it to an instance in Lightsail](#).

Step 3: Visit your Nginx instance welcome page

Navigate to the public IP address of your instance to access the application installed on it, access phpMyAdmin, or access the Bitnami documentation.

1. On your instance management page, under the **Connect** tab, make note of the public IP.
2. Browse to the public IP address, for example by going to `http://192.0.2.3`.

For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

Step 4: Map your domain name to your Nginx instance

To map your domain name, such as `example.com`, to your instance, you add a record to the domain name system (DNS) of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Networking** tab, choose **Create DNS zone**, then follow the instructions on the page.

For more information, see [Create a DNS zone to manage your domain's DNS records](#).

Step 5: Read the Bitnami documentation

Read the Bitnami documentation to learn how to deploy your Nginx application, enable HTTPS support with SSL certificates, upload files to the server with SFTP, and more.

For more information, see the [Bitnami Nginx for AWS Cloud](#).

Step 6: Create a snapshot of your Nginx instance

A snapshot is a copy of the system disk and original configuration of an instance. The snapshot includes such information as memory, CPU, disk size, and data transfer rate. You can use a snapshot as a baseline for new instances, or as a data backup.

Under the **Snapshot** tab of your instance's management page, enter a name for the snapshot, then choose **Create snapshot**.

For more information, see [Create a snapshot of your Linux or Unix instance](#).

Quick start guide: Node.js

Here are a few steps you should take to get started after your Node.js instance is up and running on Amazon Lightsail:

Step 1: Get the default application password for your Node.js instance

You need the default application password to access pre-installed applications or services on your instance.

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.
2. After you're connected, enter the following command to get the default application password:


```
cat bitnami_application_password
```

Note

If you're in a directory other than the user home directory, then enter `cat $HOME/bitnami_application_password`.

You should see a response similar to this, which contains the default application password:

```
bitnami@ip-172-31-10-100:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-172-31-10-100:~$
```



For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

Step 2: Attach a static IP address to your Node.js instance

The default dynamic public IP address attached to your instance changes every time you stop and start the instance. Create a static IP address, and attach it to your instance, to keep the public IP address from changing. Later, when you use your domain name with your instance, you don't have to update your domain's DNS records each time you stop and start the instance. You can attach one static IP to an instance.

On your instance management page, under the **Domains & DNS** tab, choose **Create static IP**, then follow the instructions on the page.

For more information, see [Create a static IP and attach it to an instance in Lightsail](#).

Step 3: Visit your Node.js instance welcome page

Navigate to the public IP address of your instance to access the application installed on it, access phpMyAdmin, or access the Bitnami documentation.

1. On your instance management page, under the **Connect** tab, make note of the public IP.
2. Browse to the public IP address, for example by going to `http://192.0.2.3`.

For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

Step 4: Map your domain name to your Node.js instance

To map your domain name, such as `example.com`, to your instance, you add a record to the domain name system (DNS) of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Networking** tab, choose **Create DNS zone**, then follow the instructions on the page.

For more information, see [Create a DNS zone to manage your domain's DNS records](#).

Step 5: Read the Bitnami documentation

Read the Bitnami documentation to learn how to deploy your Node.js application, enable HTTPS support with SSL certificates, upload files to the server with SFTP, and more.

For more information, see the [Bitnami Node.js for AWS Cloud](#).

Step 6: Create a snapshot of your Node.js instance

A snapshot is a copy of the system disk and original configuration of an instance. The snapshot includes such information as memory, CPU, disk size, and data transfer rate. You can use a snapshot as a baseline for new instances, or as a data backup.

Under the **Snapshot** tab of your instance's management page, enter a name for the snapshot, then choose **Create snapshot**.

For more information, see [Create a snapshot of your Linux or Unix instance](#).

Quick start guide: Plesk

Here are a few steps you should take to get started after your Plesk instance is up and running on Amazon Lightsail:

Important

If you experience issues after launching your Plesk instance, go to the Plesk support page to see if there are updates that need to be installed on the instance. For more information, see the [Plesk help center](#) and [Plesk Updates](#) in the *Plesk Documentation and Help Portal*.

Step 1: Get the one-time login URL for your Plesk instance

You need the one-time login URL to access the Plesk panel as an administrator.

Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.
2. After you're connected, enter the following command to get the one-time login URL:

```
sudo plesk login | grep -v internal:8
```

You should see a response similar to the following example, which contains the one-time login URL:

```
ubuntu@ip-10.10.10.10:~$ sudo plesk login
https://10.10.10.10.us-west-2.compute.amazonaws.com/login?secret=VFmhiq5NSN81d-Ebn
https://10.10.10.10/login?secret=VFmhiq5NSN81d-Ebn
ubuntu@ip-10.10.10.10:~$
```

Important

If you recently attached a static IP to your Plesk instance, you might get a one-time login URL that uses the old public IP address. Reboot the instance, and then run the above command again to get a one-time login URL that uses the new static, public IP address.

3. Copy the URL to your clipboard, or make note of it. You will need it later to sign in to the Plesk panel for the first time.

For more information, see [Set up and configure Plesk on Lightsail](#).

Step 2: Sign in to the Plesk panel for the first time

Paste the one-time login URL into a web browser. Follow the instructions on the page to create your sign in credentials for Plesk. You should see an option to add your domain to Plesk when you sign in for the first time.

Note

You might see a browser warning that your connection is not private, not secure, or that there's a security risk. This happens because your Plesk instance does not yet have an SSL/TLS certificate applied to it. In the browser window, choose **Advanced**, **Details**, or **More**

information to view the options that are available. Then choose to proceed to the website even if it's not private or secure.

For more information, see [Set up and configure Plesk on Lightsail](#).

Step 3: Attach a static IP address to your Plesk instance

The default dynamic public IP address attached to your instance changes every time you stop and start the instance. Create a static IP address, and attach it to your instance, to keep the public IP address from changing. Later, when you use your domain name with your instance, you don't have to update your domain's DNS records each time you stop and start the instance. You can attach one static IP to an instance.

On your instance management page, under the **Networking** tab, choose **Create static IP**, then follow the instructions on the page.

For more information, see [Create a static IP and attach it to an instance](#).

Step 4: Map your domain name to your Plesk instance

Note

You can map a domain to your Plesk instance, which you can use to access your Plesk panel. You can also map multiple domains within the Plesk panel, which you can use to manage websites within the Plesk panel. This section describes how to map your domain to your Plesk instance. For more information about mapping multiple domains within the Plesk panel, see [Adding a Domain in Plesk](#) in the *Plesk Documentation and Help Portal*.

To map your domain name, such as `example.com`, to your instance, you add a record to the domain name system (DNS) of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page.

For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

Step 5: Read the Plesk documentation

Read the Plesk documentation to learn how to administer web sites using Plesk, customize the Plesk panel, and more.

For more information, see the [Getting Started with Managing Websites in Plesk](#) in the *Plesk Documentation and Help Portal*.

Step 6: Create a snapshot of your Plesk instance

A snapshot is a copy of the system disk and original configuration of an instance. The snapshot includes such information as memory, CPU, disk size, and data transfer rate. You can use a snapshot as a baseline for new instances, or as a data backup.

Under the **Snapshot** tab of your instance's management page, enter a name for the snapshot, then choose **Create snapshot**.

For more information, see [Create a snapshot of your Linux or Unix instance](#).

Quick start guide: PrestaShop

Here are a few steps you should complete to get started after your PrestaShop instance is up and running on Amazon Lightsail.

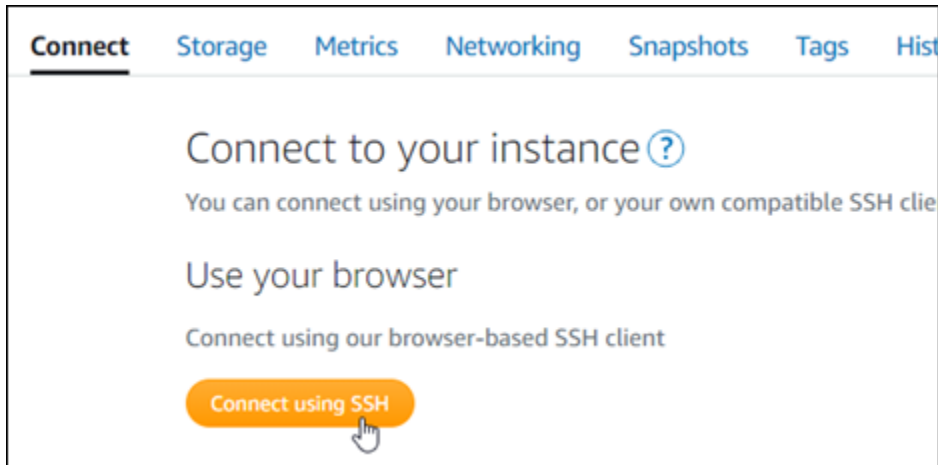
Contents

- [Step 1: Get the default application password for your PrestaShop website](#)
- [Step 2: Attach a static IP address to your PrestaShop instance](#)
- [Step 3: Sign in to the administration dashboard of your PrestaShop website](#)
- [Step 4: Route traffic for your registered domain name to your PrestaShop website](#)
- [Step 5: Configure HTTPS for your PrestaShop website](#)
- [Step 6: Configure SMTP for email notifications](#)
- [Step 7: Read the Bitnami and PrestaShop documentation](#)
- [Step 8: Create a snapshot of your PrestaShop instance](#)

Step 1: Get the default application password for your PrestaShop website

Complete the following steps to get the default application password for your PrestaShop website.

1. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to get the default application password:

```
cat $HOME/bitnami_application_password
```

You should see a response similar to the following example, which contains the default application password. Store this password in a safe place. You will use it in the next section of this tutorial to sign in to the administration dashboard of your PrestaShop website.

```
bitnami@ip-172-31-10-100:~$ cat bitnami_application_password
JeVN8xDwLCIp
bitnami@ip-172-31-10-100:~$
```

For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

Step 2: Attach a static IP address to your PrestaShop instance

The public IP address assigned to your instance when you first create it will change every time you stop and start your instance. You should create and attach a static IP address to your instance to ensure its public IP address doesn't change. Later, when you use a registered domain name, such as example.com, with your instance, you don't have to update your domain's DNS records every time you stop and start your instance. You can attach one static IP to an instance.

On the instance management page, under the **Networking** tab, choose **Create a static IP** or **Attach static IP** (if you previously created a static IP that you can attach to your instance), then follow the instructions on the page.



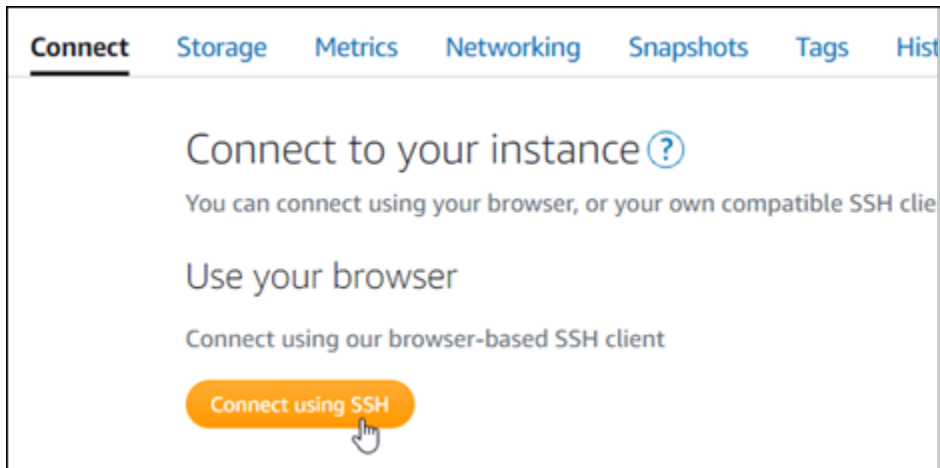
For more information, see [Create a static IP and attach it to an instance](#).

After the new static IP address is attached to your instance, you must complete the following steps to make the PrestaShop software aware of the new static IP address.

1. Make a note of the static IP address of your instance. It's listed in the header section of your instance management page.



2. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



3. After you're connected, enter the following command. Be sure to replace *<StaticIP>* with the new static IP address of your instance.

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

You should see a response similar to the following example. The PrestaShop software should now be aware of the new static IP address.

```
bitnami@ip-173-36-0-197:~$ sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
Configuring domain to 203.0.113.0
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO ==> Updating hostname in database
prestashop 15:49:22.46 INFO ==> Purging cache
Disabling automatic domain update for IP address changes
```

Note

PrestaShop does not currently support IPv6 addresses. You can enable IPv6 for the instance, but the PrestaShop software will not respond to requests over the IPv6 network.

Step 3: Sign in to the administration dashboard of your PrestaShop website

Complete the following step to access your PrestaShop website and sign in to its administration dashboard. To sign in, you will use the default user name (`user@example.com`) and the default application password that you got earlier in this guide.

1. In the Lightsail console, make note of the public or static IP address that is listed in the header area of the instance management page.



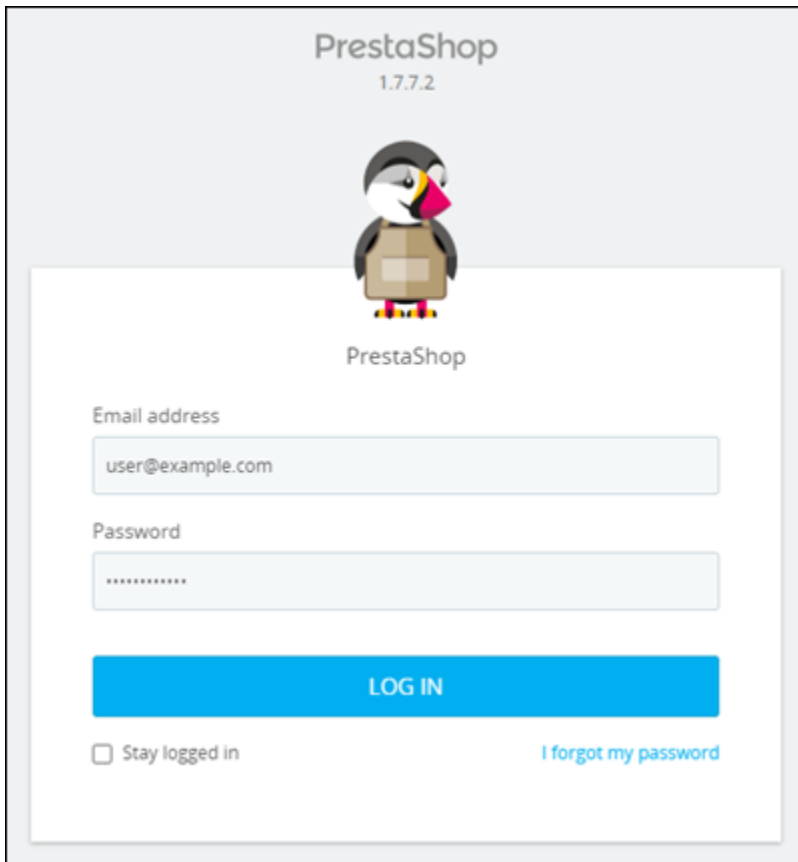
2. Browse to the following address to access the sign in page for the administration dashboard of your PrestaShop website. Be sure to replace `<InstanceIpAddress>` with the public or static IP address of your instance.

```
http://<InstanceIpAddress>/administration
```

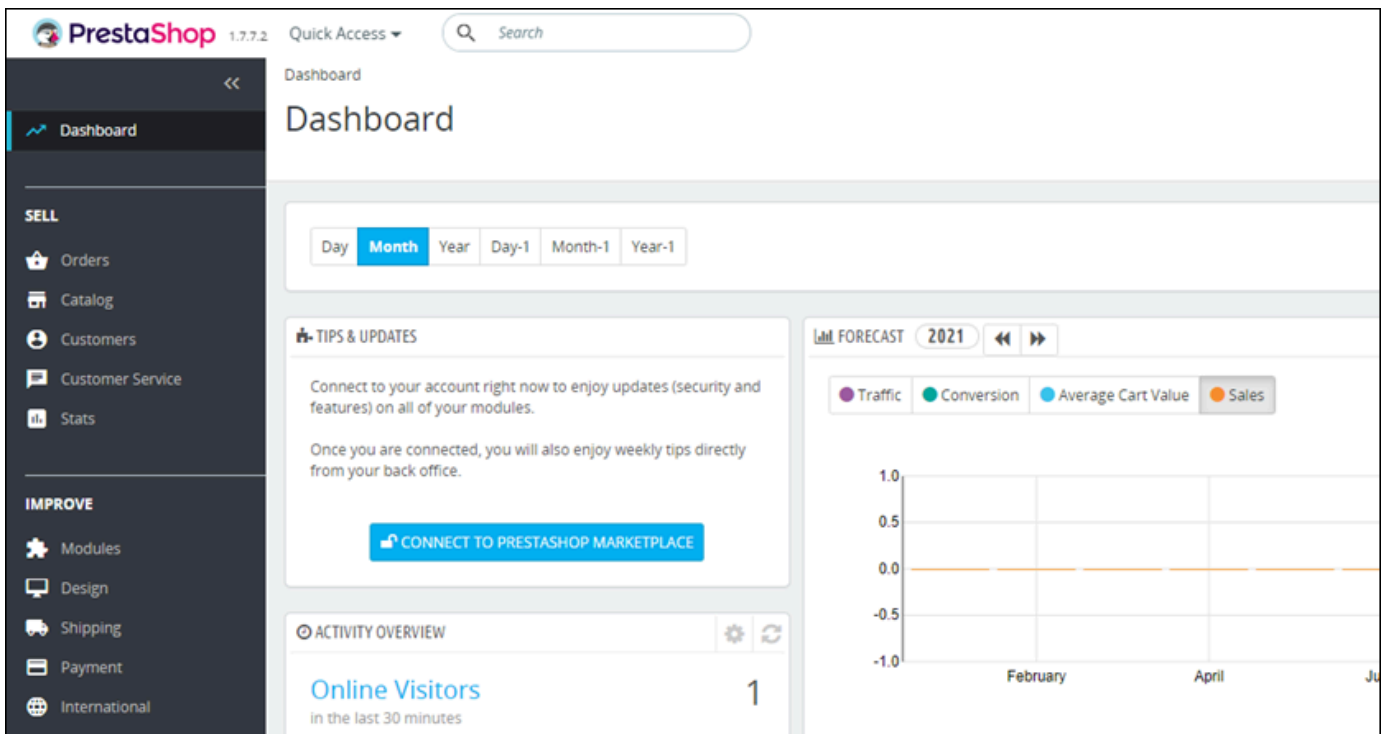
Example:

```
http://203.0.113.0/administration
```

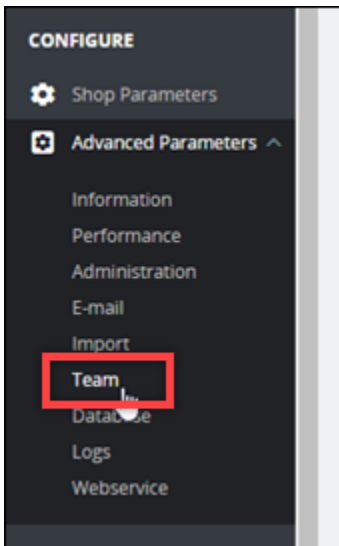
3. Enter the default user name (`user@example.com`), the default application password you got earlier in this guide, and choose **Log in**.



The PrestaShop administration dashboard appears.



To change the default user name or password that you use to sign in to the administration dashboard of your PrestaShop website, choose **Advanced Parameters** in the navigation pane, and then choose **Team**. For more information, see [User Guide PrestaShop](#) in the *PrestaShop documentation*.



For more information about the administration dashboard, see For more information, see [User Guide PrestaShop](#) in the *PrestaShop documentation*.

Step 4: Route traffic for your registered domain name to your PrestaShop website

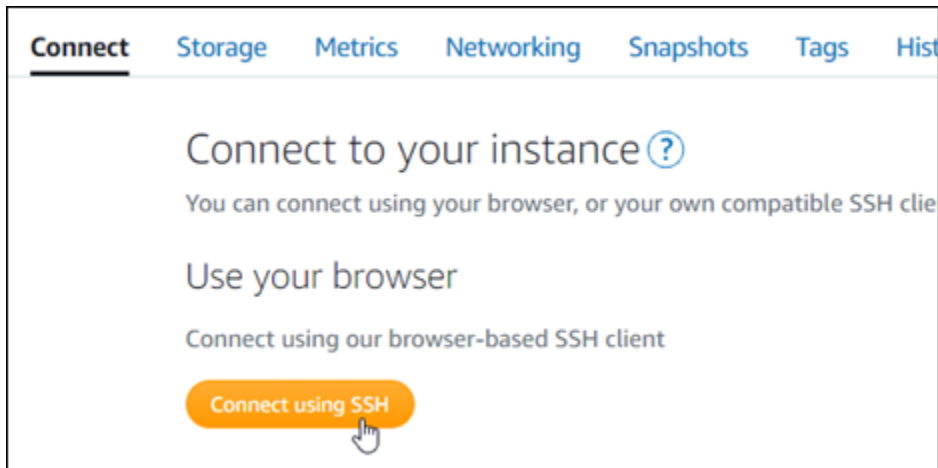
To route traffic for your registered domain name, such as `example.com`, to your PrestaShop website, you add a record to the domain name system (DNS) of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page.

For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

After your domain name is routing traffic to your instance, you must complete the following steps to make the PrestaShop software aware of the domain name.

1. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command. Be sure to replace *<DomainName>* with the domain name that is routing traffic to your instance.

```
sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain www.example.com
```

You should see a response similar to the following example. The PrestaShop software should now be aware of the domain name.

```
bitnami@ip-173-20-0-100:~$ sudo /opt/bitnami/configure_app_domain --domain www.example.com
Configuring domain to www.example.com
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO ==> Updating hostname in database
prestashop 15:49:22.46 INFO ==> Purging cache
Disabling automatic domain update for IP address changes
```

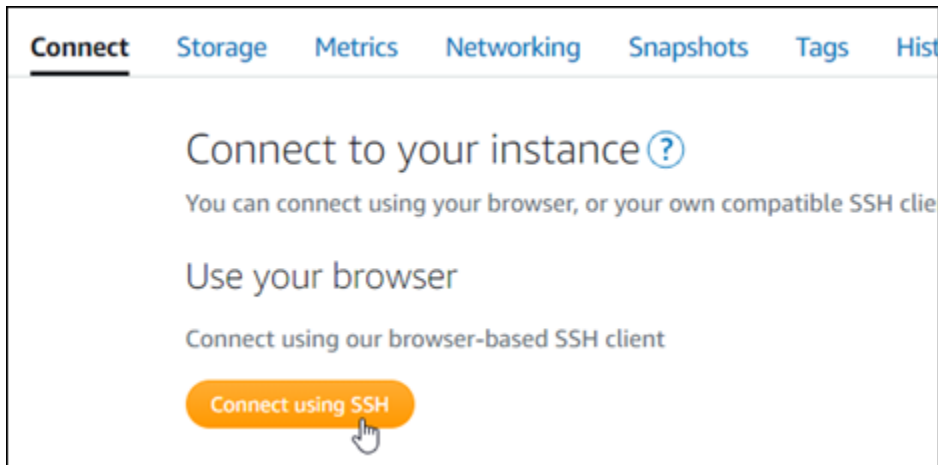
Step 5: Configure HTTPS for your PrestaShop website

Complete the following steps to configure HTTPS on your PrestaShop website. These steps show you how to use the Bitnami HTTPS configuration tool (bncert), which is a command line tool for requesting SSL/TLS certificates, setting up redirections (e.g. HTTP to HTTPS), and renewing certificates.

⚠ Important

The bncert tool will issue certificates only for domains that are currently routing traffic to the public IP address of your PrestaShop instance. Before starting with these steps, make sure that you add DNS records to the DNS of all domains that you want to use with your PrestaShop website.

1. On the instance management page, under the Connect tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to start the bncert-tool.

```
sudo /opt/bitnami/bncert-tool
```

You should see a response similar to the following example:

```
bitnami@ip-172-31-7-10:~$ sudo /opt/bitnami/bncert-tool
-----
Welcome to the Bitnami HTTPS Configuration tool.
-----
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: █
```

3. Enter your primary domain name and alternate domain names separated by a space as shown in the following example.

```
-----  
Welcome to the Bitnami HTTPS Configuration tool.  
-----  
Domains  
Please provide a valid space-separated list of domains for which you wish to  
configure your web server.  
Domain list []: example.com www.example.com
```

4. The bncert tool will ask how you want your website's redirection to be configured. These are the options available:
 - **Enable HTTP to HTTPS redirection** - Specifies whether users who browse to the HTTP version of your website (i.e., `http://example.com`) are automatically redirected to the HTTPS version (i.e., `https://example.com`). We recommend enabling this option because it forces all visitors to use the encrypted connection. Type Y and press **Enter** to enable it.
 - **Enable non-www to www redirection** - Specifies whether users who browse to the apex of your domain (i.e., `https://example.com`) are automatically redirected to your domain's www subdomain (i.e., `https://www.example.com`). We recommend enabling this option. However, you may want to disable it and enable the alternate option (enable www to non-www redirection) if you have specified the apex of your domain as your preferred website address in search engine tools like Google's webmaster tools, or if your apex points directly to your IP and your www subdomain references your apex via a CNAME record. Type Y and press **Enter** to enable it.
 - **Enable www to non-www redirection** - Specifies whether users who browse to your domain's www subdomain (i.e., `https://www.example.com`) are automatically redirected to the apex of your domain (i.e., `https://example.com`). We recommend disabling this, if you enabled non-www redirection to www. Type N and press **Enter** to disable it.

Your selections should look like the following example.


```
Enable/disable redirections

Please select the redirections you wish to enable or disable on your Bitnami
installation.

Enable HTTP to HTTPS redirection [Y/n]: Y

Enable non-www to www redirection [Y/n]: Y

Enable www to non-www redirection [y/N]: N
```

5. The changes that are going to be made are listed. Type Y and press **Enter** to confirm and continue.

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
www.example.com)
7. Start web server once all changes have been performed

Do you agree to these changes? [Y/n]: Y
```

6. Enter your email address to associate with your Let's Encrypt certificate and press **Enter**.

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []: 
```

7. Review the Let's Encrypt Subscriber Agreement. Type Y and press **Enter** to accept the agreement and continue.

```
The Let's Encrypt Subscriber Agreement can be found at:  
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf  
Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]: █
```

The actions are performed to enable HTTPS on your instance, including requesting the certificate and configuring the redirections you specified.

```
Performing changes to your installation  
  
The Bitnami HTTPS Configuration Tool will perform any necessary actions to your  
Bitnami installation. This may take some time, please be patient.  
  
█
```

Your certificate is successfully issued and validated, and the redirections are successfully configured on your instance if you see a message similar to the following example.

```
Success  
  
The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.  
The configuration report is shown below.  
  
Backup files:  
* /opt/bitnami/apache2/conf/httpd.conf.back.202005290035  
* /opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf.back.202005290035  
* /opt/bitnami/apache2/conf/bitnami/bitnami.conf.back.202005290035  
  
Find more details in the log file:  
  
/tmp/bncert-202005290035.log  
  
If you find any issues, please check Bitnami Support forums at:  
  
https://community.bitnami.com  
  
Press [Enter] to continue: █
```

The bncert tool will perform an automatic renewal of your certificate every 80 days before it expires. Continue to the next set of steps to finish enabling HTTPS on your PrestaShop website.

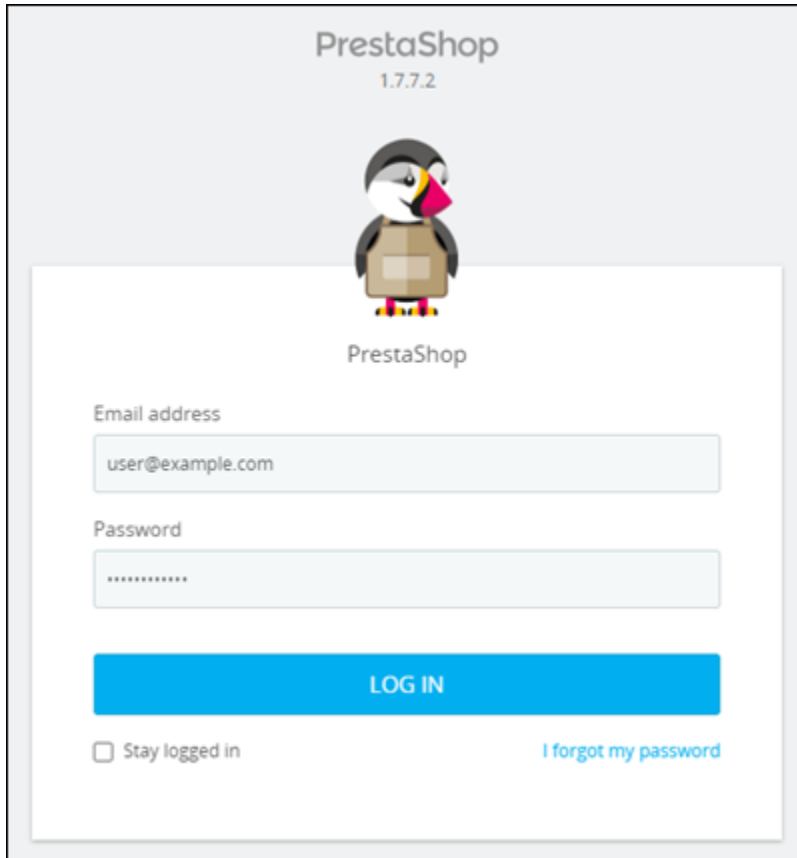
8. Browse to the following address to access the sign in page for the administration dashboard of your PrestaShop website. Be sure to replace *<DomainName>* with the registered domain name that is routing traffic to your instance.

```
http://<DomainName>/administration
```

Example:

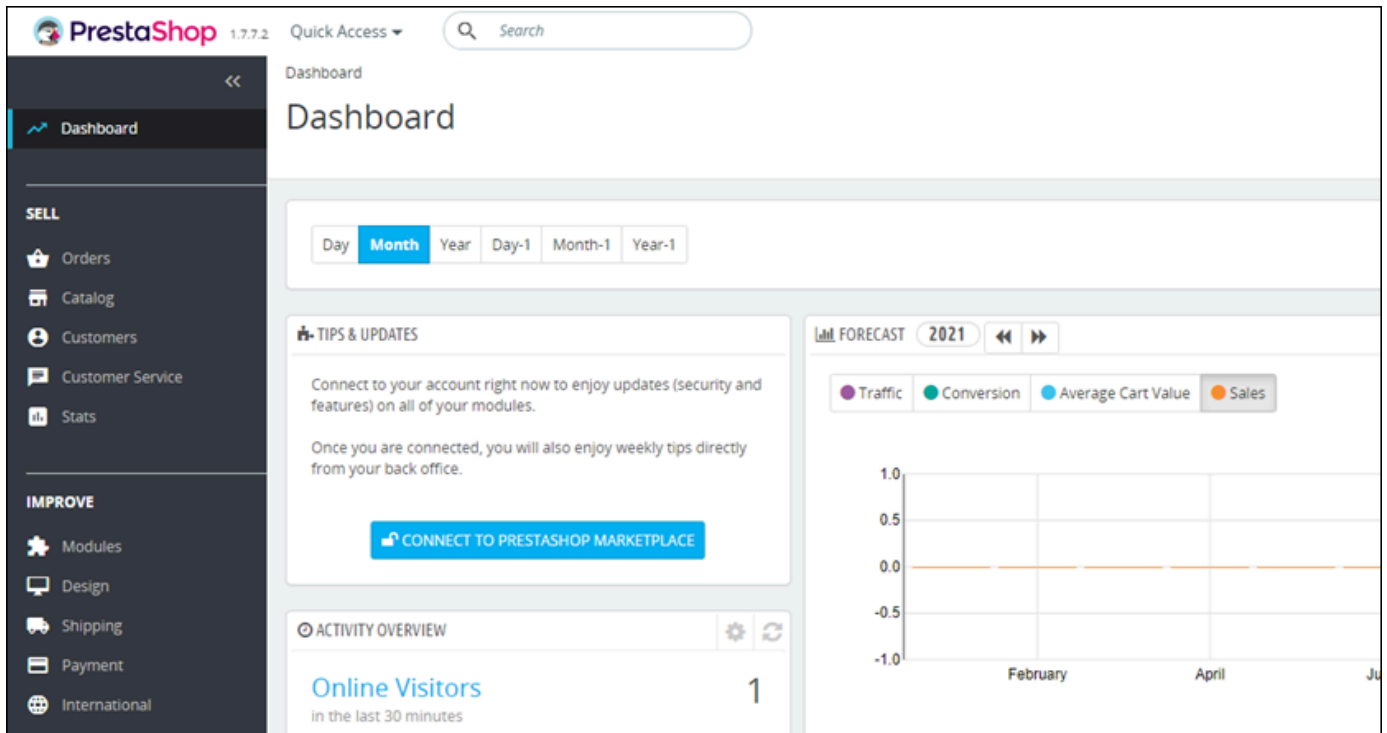
`http://www.example.com/administration`

9. Enter the default user name (user@example.com), the default application password you got earlier in this guide, and choose **Log in**.

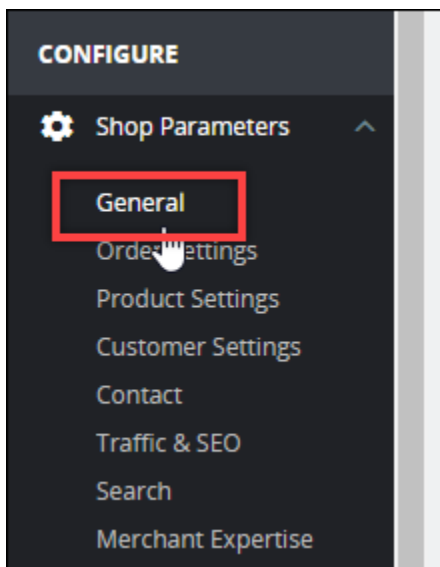


The screenshot shows the PrestaShop 1.7.7.2 administration login interface. At the top, it displays 'PrestaShop 1.7.7.2' and a cartoon penguin mascot. Below the mascot is the 'PrestaShop' logo. The login form consists of two input fields: 'Email address' with the value 'user@example.com' and 'Password' with masked characters. A prominent blue 'LOG IN' button is centered below the fields. At the bottom left, there is a checkbox for 'Stay logged in', and at the bottom right, there is a link for 'I forgot my password'.

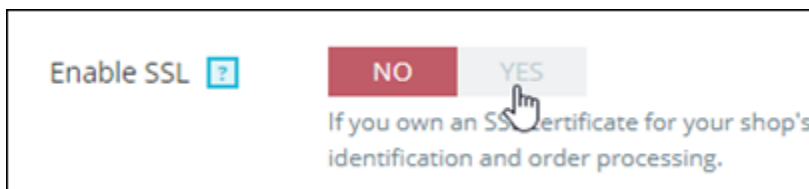
The PrestaShop administration dashboard appears.



10. Choose **Shop Parameters** in the navigation pane, and then choose **General**.



11. Choose **Yes** next to **Enable SSL**.



12. Scroll to the bottom of the page and choose **Save**.

13. When the **General** page reloads, choose **Yes** next to **Enable SSL on all pages**.

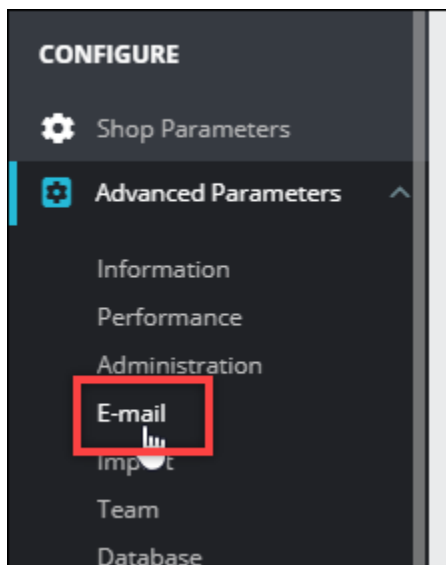


14. Scroll to the bottom of the page and choose **Save**.

HTTPS is now configured for your PrestaShop website. When customers browse to the HTTP version (e.g., `http://www.example.com`) of your PrestaShop website, they will be automatically redirected to the HTTPS version (e.g., `https://www.example.com`).

Step 6: Configure SMTP for email notifications

Configure the SMTP settings of your PrestaShop website to enable email notifications for it. To do so, sign in to the administration dashboard of your PrestaShop website. Choose **Advanced Parameters** in the navigation pane, and then choose **E-mail**. You should also adjust your email contacts accordingly. To do so, choose **Shop Parameters** in the navigation pane, and then choose **Contact**.



For more information, For more information, see [User Guide PrestaShop](#) in the *PrestaShop documentation* and [Configure SMTP for outbound emails](#) in the Bitnami documentation.

Important

If you configure SMTP to use ports 25, 465, or 587, then you must open those ports in the firewall of your instance in the Lightsail console. For more information, see [Adding and editing instance firewall rules in Amazon Lightsail](#).

If you configure your Gmail account to send email on your PrestaShop website, then you must use an app password instead of using the standard password that you use to sign in to Gmail. For more information, see [Sign in with App Passwords](#).

Step 7: Read the Bitnami and PrestaShop documentation

Read the Bitnami documentation to learn how to perform administrative tasks on your PrestaShop instance and website, such as install plugins and customize the theme. For more information, see [Bitnami PrestaShop Stack for AWS Cloud](#) in the *Bitnami documentation*.

You should also read the PrestaShop documentation to learn how to administer your PrestaShop website. For more information, see the [User Guide PrestaShop](#) in the *PrestaShop documentation*.

Step 8: Create a snapshot of your PrestaShop instance

After you configure your PrestaShop website the way you want it, create periodic snapshots of your instance to back it up. You can create snapshots manually, or enable automatic snapshots to have Lightsail create daily snapshots for you. If something goes wrong with your instance, you can create a new replacement instance using the snapshot. For more information, see [Snapshots](#).

On the instance management page, under the **Snapshot** tab, choose **Create a snapshot** or choose to enable automatic snapshots.

Connect
Storage
Metrics
Networking
Snapshots
Tags
History
Delete

Manual snapshots ?

You can create a snapshot to back up your instance, its system disk, and attached disks.

[+ Create snapshot](#)

> February 5, 2021 - 9:37 AM	"Prestashop-1612546662"	⋮
> January 13, 2021 - 9:44 AM	"Prestashop-1610559880"	⋮
> December 9, 2020 - 12:33 PM	"Prestashop-1607545986"	⋮
> September 9, 2020 - 5:44 PM	"Prestashop-1599698658"	⋮

Showing 4 of 4 snapshots

Automatic snapshots ?

Automatic snapshots are enabled

Your daily snapshot time is 10:00 PM PST.
We will store your seven most recent snapshots.

[Change snapshot time](#)

DAILY SNAPSHOTS

> Thursday	March 4, 2021	⋮
> Wednesday	March 3, 2021	⋮
> Tuesday	March 2, 2021	⋮

For more information, see [Creating a snapshot of your Linux or Unix instance in Amazon Lightsail](#) or [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

Quick start guide: Redmine

Here are a few steps you should take to get started after your Redmine instance is up and running on Amazon Lightsail:

Contents

- [Step 1: Read the Bitnami documentation](#)
- [Step 2: Get the default application password to access the Redmine administration dashboard](#)

- [Step 3: Attach a static IP address to your instance](#)
- [Step 4: Sign in to the administration dashboard of your Redmine website](#)
- [Step 5: Route traffic for your registered domain name to your Redmine website](#)
- [Step 6: Configure HTTPS for your Redmine website](#)
- [Step 7: Read the Redmine documentation and continue configuring your website](#)
- [Step 8: Create a snapshot of your instance](#)

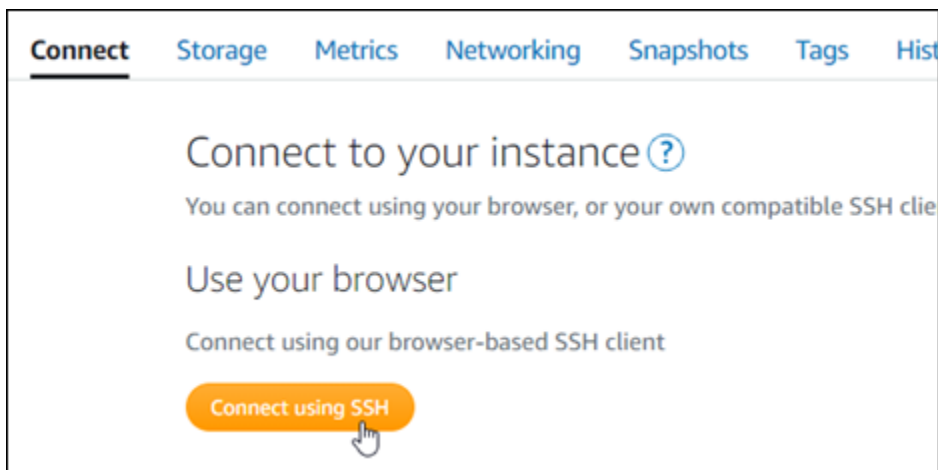
Step 1: Read the Bitnami documentation

Read the Bitnami documentation to learn how to configure your Redmine application. For more information, see the [Redmine Packaged By Bitnami For AWS Cloud](#).

Step 2: Get the default application password to access the Redmine administration dashboard

Complete the following procedure to get the default application password required to access the administration dashboard for your Redmine website. For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.

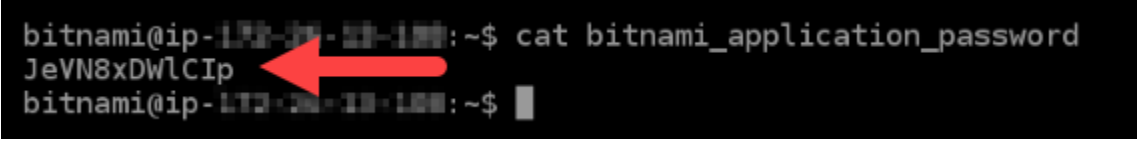


2. After you're connected, enter the following command to get the application password:

```
cat $HOME/bitnami_application_password
```

You should see a response similar to the following example, which contains the default application password:


```
bitnami@ip-192-0-2-0:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-192-0-2-0:~$
```



Step 3: Attach a static IP address to your instance

The public IP address assigned to your instance when you first create it will change every time you stop and start your instance. You should create and attach a static IP address to your instance to ensure its public IP address doesn't change. Later, when you use a registered domain name, such as `example.com`, with your instance, you don't have to update your domain's DNS records every time you stop and start your instance. You can attach one static IP to an instance.

On the instance management page, under the **Networking** tab, choose **Create a static IP** or **Attach static IP** (if you previously created a static IP that you can attach to your instance), then follow the instructions on the page. For more information, see [Create a static IP and attach it to an instance](#).

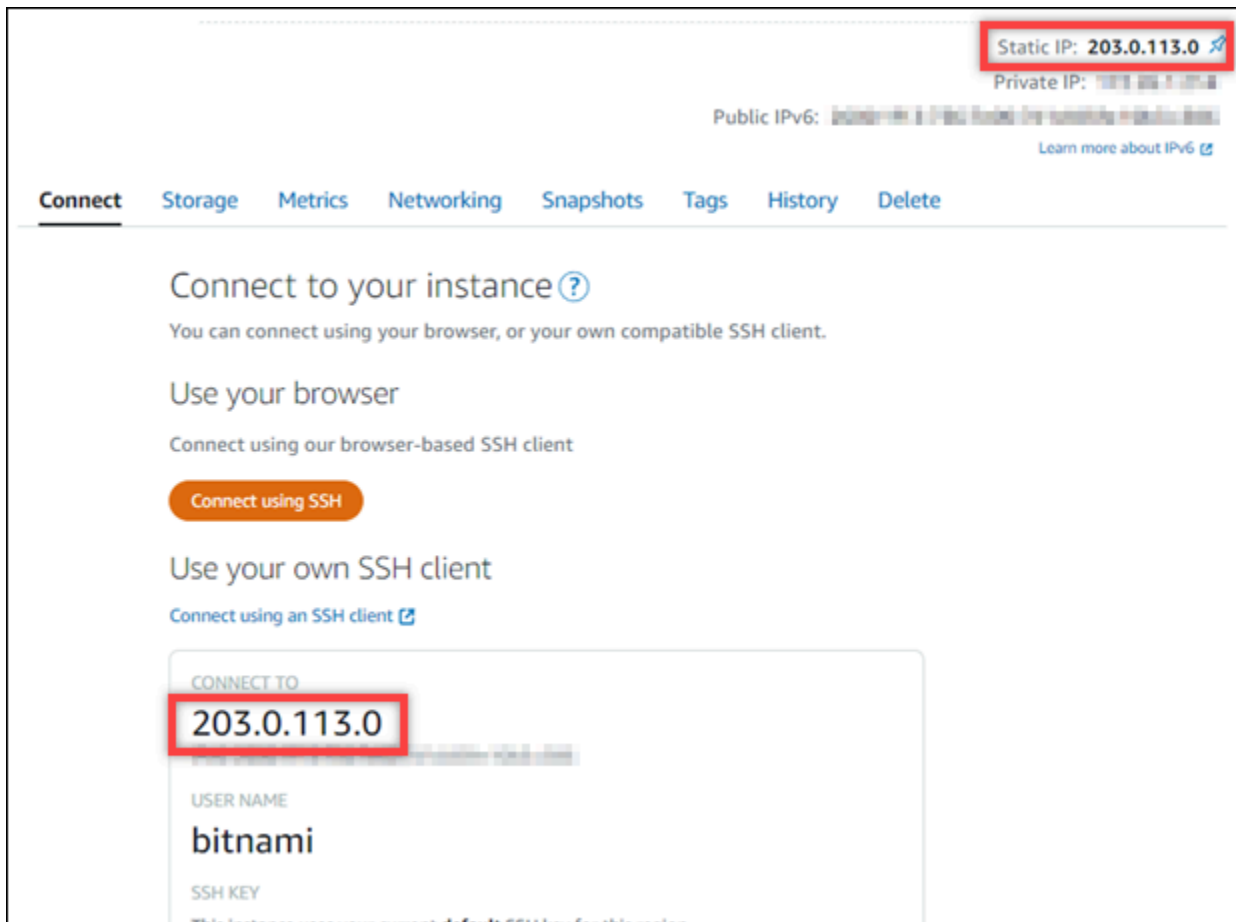


The screenshot shows the 'Networking' tab in the AWS Lightsail console. Under 'IPv4 networking', there is a section for 'PUBLIC IP' showing the address '192.0.2.0'. Below the address is a button labeled '+ Create static IP' with a red arrow pointing to it. To the right, there is a 'PRIVATE IP' section showing '172.31.0.1' and a 'What's new?' link. Below the public IP section, there is a note: 'Your public IPv4 address changes when you stop and start your instance. Attach a static IPv4 address to your instance to keep it from changing.'

Step 4: Sign in to the administration dashboard of your Redmine website

Now that you have the default application password, complete the following procedure to navigate to your Redmine website's home page, and sign in to the administration dashboard. After you're signed in, you can start customizing your website and making administrative changes. For more information about what you can do in Joomla!, see the [Step 7: Read the Redmine documentation and continue configuring your website](#) section later in this guide.

1. On your instance management page, under the **Connect** tab, make note of the public IP address of your instance. The public IP address is also displayed in the header section of your instance management page.



2. Browse to the public IP address of your instance, for example by going to `http://203.0.113.0`.

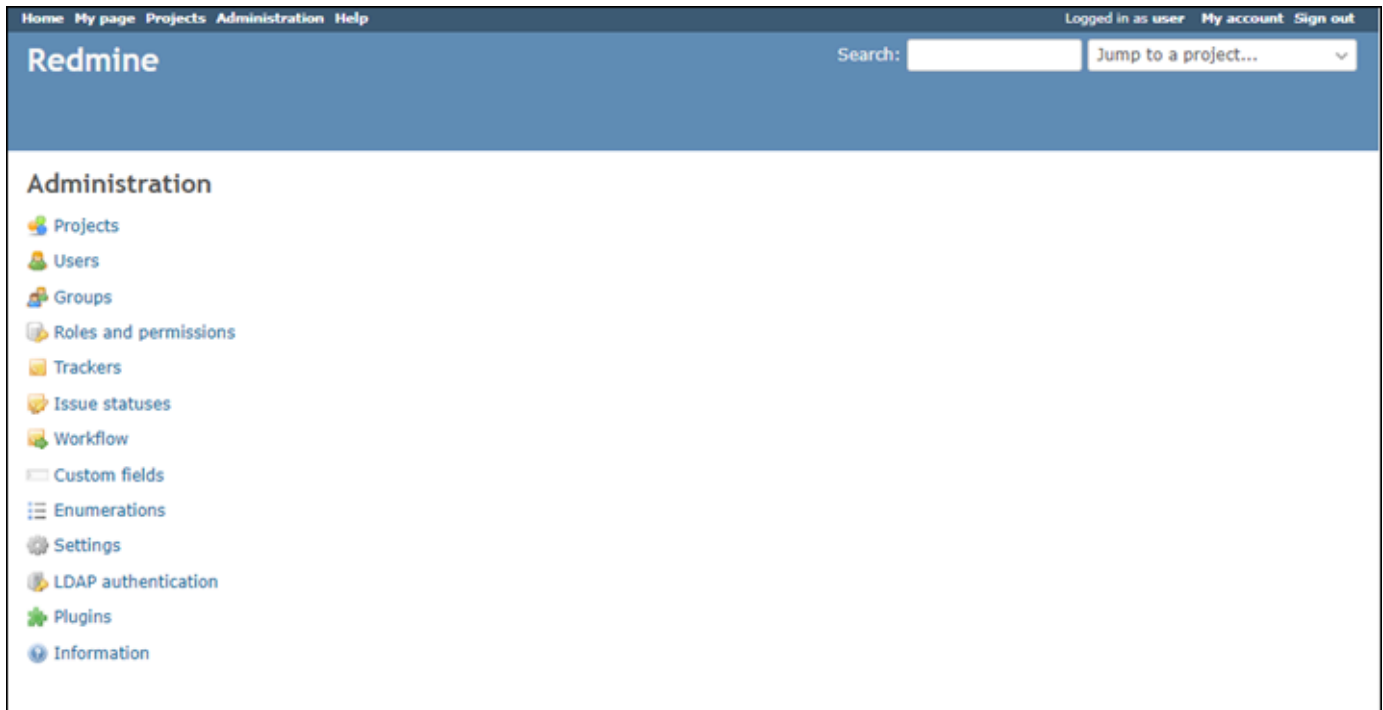
The home page of your Redmine website should appear.

3. Choose **Manage** in the bottom right corner of your Redmine website home page.

If the **Manage** banner is not shown, you can reach the sign in page by browsing to `http://<PublicIP>/admin`. Replace `<PublicIP>` with the public IP address of your instance.

4. Sign in using the default user name (`user1`) and the default password retrieved earlier in this guide.

The Redmine administration dashboard appears.



Step 5: Route traffic for your registered domain name to your Redmine website

To route traffic for your registered domain name, such as `example.com`, to your Redmine website, you add a record to the DNS of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page. For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

If you browse to the domain name that you configured for your instance, you should be redirected to the home page of your Redmine website. Next, you should generate and configure an SSL/TLS certificate to enable HTTPS connections for your Redmine website. For more information, continue to the next [Step 6: Configure HTTPS for your Redmine website](#) section of this guide.

Step 6: Configure HTTPS for your Redmine website

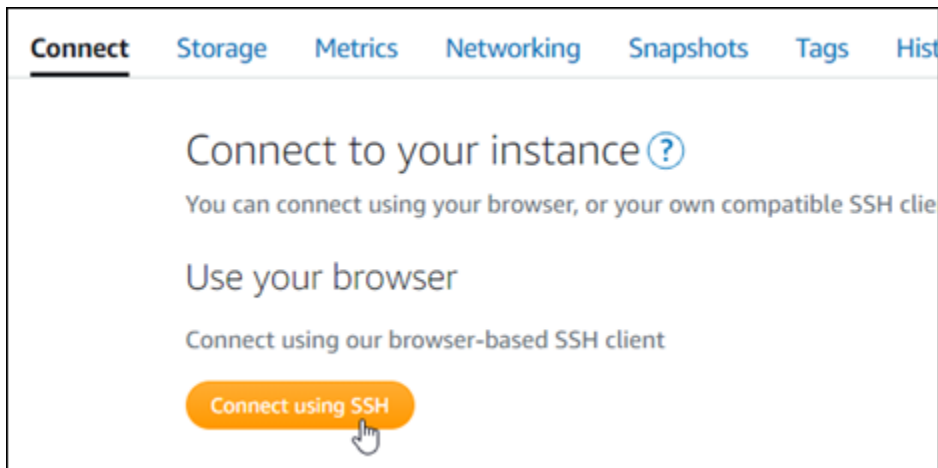
Complete the following procedure to configure HTTPS on your Redmine website. These steps show you how to use the Bitnami HTTPS Configuration Tool (`bncert-tool`), which is a command line

tool for requesting Let's Encrypt SSL/TLS certificates. For more information see [Learn About The Bitnami HTTPS Configuration Tool](#) in the *Bitnami documentation*.

⚠ Important

Before starting with this procedure, make sure that you configured your domain to route traffic to your Redmine instance. Otherwise, the SSL/TLS certificate validation process will fail.

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to confirm the bncert tool is installed on your instance.

```
sudo /opt/bitnami/bncert-tool
```

You should see one of the following responses:

- If you see command not found in the response, then the bncert tool is not installed on your instance. Continue to the next step in this procedure to install the bncert tool on your instance.
- If you see **Welcome to the Bitnami HTTPS configuration tool** in the response, then the bncert tool is installed on your instance. Continue to the step 8 of this procedure.
- If the bncert tool has been installed on your instance for a while, then you might see a message indicating that an updated version of the tool is available. Choose to download it, and then enter the `sudo /opt/bitnami/bncert-tool` command to run the bncert tool again. Continue to the step 8 of this procedure.

3. Enter the following command to download the bncert run file to your instance.

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/bncert-linux-x64.run
```

4. Enter the following command to create a directory for the bncert tool run file on your instance.

```
sudo mkdir /opt/bitnami/bncert
```

5. Enter the following command to make the bncert run a file that can be executed as a program.

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. Enter the following command to create a symbolic link that runs the bncert tool when you enter the `sudo /opt/bitnami/bncert-tool` command.

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

You are now done installing the bncert tool on your instance.

7. Enter the following command to run the bncert tool.

```
sudo /opt/bitnami/bncert-tool
```

8. Enter your primary domain name and alternate domain names separated by a space as shown in the following example.

If your domain is not configured to route traffic to the public IP address of your instance, the bncert tool will ask you to make that configuration before continuing. Your domain must be routing traffic to the public IP address of the instance from which you are using the bncert tool to enable HTTPS on the instance. This confirms that you own the domain, and serves as the validation for your certificate.

```
-----
Welcome to the Bitnami HTTPS Configuration tool.
-----
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: example.com www.example.com
```

9. The `bncert` tool will ask you how you want your website's redirection to be configured. These are the options available:
- **Enable HTTP to HTTPS redirection** - Specifies whether users who browse to the HTTP version of your website (i.e., `http://example.com`) are automatically redirected to the HTTPS version (i.e., `https://example.com`). We recommend enabling this option because it forces all visitors to use the encrypted connection. Type **Y** and press **Enter** to enable it.
 - **Enable non-www to www redirection** - Specifies whether users who browse to the apex of your domain (i.e., `https://example.com`) are automatically redirected to your domain's `www` subdomain (i.e., `https://www.example.com`). We recommend enabling this option. However, you may want to disable it and enable the alternate option (enable `www` to non-`www` redirection) if you have specified the apex of your domain as your preferred website address in search engine tools like Google's webmaster tools, or if your apex points directly to your IP and your `www` subdomain references your apex via a CNAME record. Type **Y** and press **Enter** to enable it.
 - **Enable www to non-www redirection** - Specifies whether users who browse to your domain's `www` subdomain (i.e., `https://www.example.com`) are automatically redirected to the apex of your domain (i.e., `https://example.com`). We recommend disabling this, if you enabled non-`www` redirection to `www`. Type **N** and press **Enter** to disable it.

Your selections should look like the following example.

```
Enable/disable redirections
Please select the redirections you wish to enable or disable on your Bitnami
installation.

Enable HTTP to HTTPS redirection [Y/n]: Y

Enable non-www to www redirection [Y/n]: Y

Enable www to non-www redirection [y/N]: N
```

10. The changes that are going to be made are listed. Type **Y** and press **Enter** to confirm and continue.

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
www.example.com)
7. Start web server once all changes have been performed

Do you agree to these changes? [Y/n]: Y
```

11. Enter your email address to associate with your Let's Encrypt certificate and press **Enter**.

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []: 
```

12. Review the Let's Encrypt Subscriber Agreement. Type Y and press **Enter** to accept the agreement and continue.

```
The Let's Encrypt Subscriber Agreement can be found at:

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]: 
```

The actions are performed to enable HTTPS on your instance, including requesting the certificate and configuring the redirections you specified.

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.

|
```

Your certificate is successfully issued and validated, and the redirections are successfully configured on your instance if you see a message similar to the following example.

```
Success

The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.

The configuration report is shown below.

Backup files:
* /opt/bitnami/apache2/conf/httpd.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami.conf.back.202005290035

Find more details in the log file:

/tmp/bncert-202005290035.log

If you find any issues, please check Bitnami Support forums at:

https://community.bitnami.com

Press [Enter] to continue:█
```

The bncert tool will perform an automatic renewal of your certificate every 80 days before it expires. Repeat the above steps if you wish to use additional domains and subdomains with your instance, and you want to enable HTTPS for those domains.

You are now done enabling HTTPS on your Redmine instance. Next time you browse to your Redmine website using the domain you configured, you should see that it redirects to the HTTPS connection.

Step 7: Read the Redmine documentation and continue configuring your website

Read the Redmine documentation to learn how to administer and customize your website. For more information, see the [Redmine guide](#).

Step 8: Create a snapshot of your instance

After you configure your Redmine website the way you want it, create periodic snapshots of your instance to back it up. You can create snapshots manually, or enable automatic snapshots to have Lightsail create daily snapshots for you. If something goes wrong with your instance, you can create a new replacement instance using the snapshot. For more information, see [Snapshots](#).

On the instance management page, under the **Snapshot** tab, choose **Create a snapshot** or choose to enable automatic snapshots.

Connect
Storage
Metrics
Networking
Snapshots
Tags
History
Delete

Manual snapshots ?

You can create a snapshot to back up your instance, its system disk, and attached disks.

[+ Create snapshot](#)

>		February 5, 2021 - 9:37 AM	"Prestashop-1612546662"	⋮
>		January 13, 2021 - 9:44 AM	"Prestashop-1610559880"	⋮
>		December 9, 2020 - 12:33 PM	"Prestashop-1607545986"	⋮
>		September 9, 2020 - 5:44 PM	"Prestashop-1599698658"	⋮

Showing 4 of 4 snapshots

Automatic snapshots ?

Automatic snapshots are enabled

Your daily snapshot time is 10:00 PM PST.
We will store your seven most recent snapshots.

[Change snapshot time](#)

DAILY SNAPSHOTS

>		Thursday	March 4, 2021	⋮
>		Wednesday	March 3, 2021	⋮
>		Tuesday	March 2, 2021	⋮

For more information, see [Creating a snapshot of your Linux or Unix instance in Amazon Lightsail](#) or [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

Quick start guide: WordPress

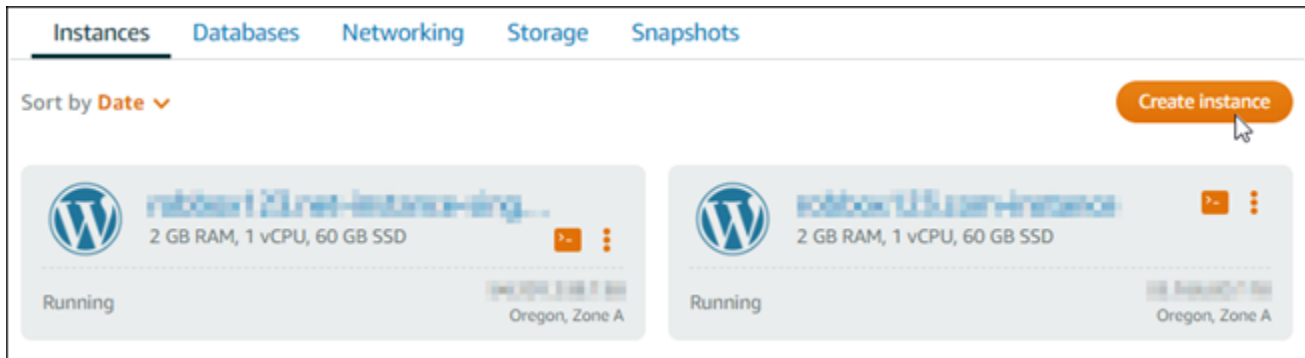
With this quick start guide, you'll learn how to launch and configure a WordPress instance on Amazon Lightsail.

Step 1: Create a WordPress instance

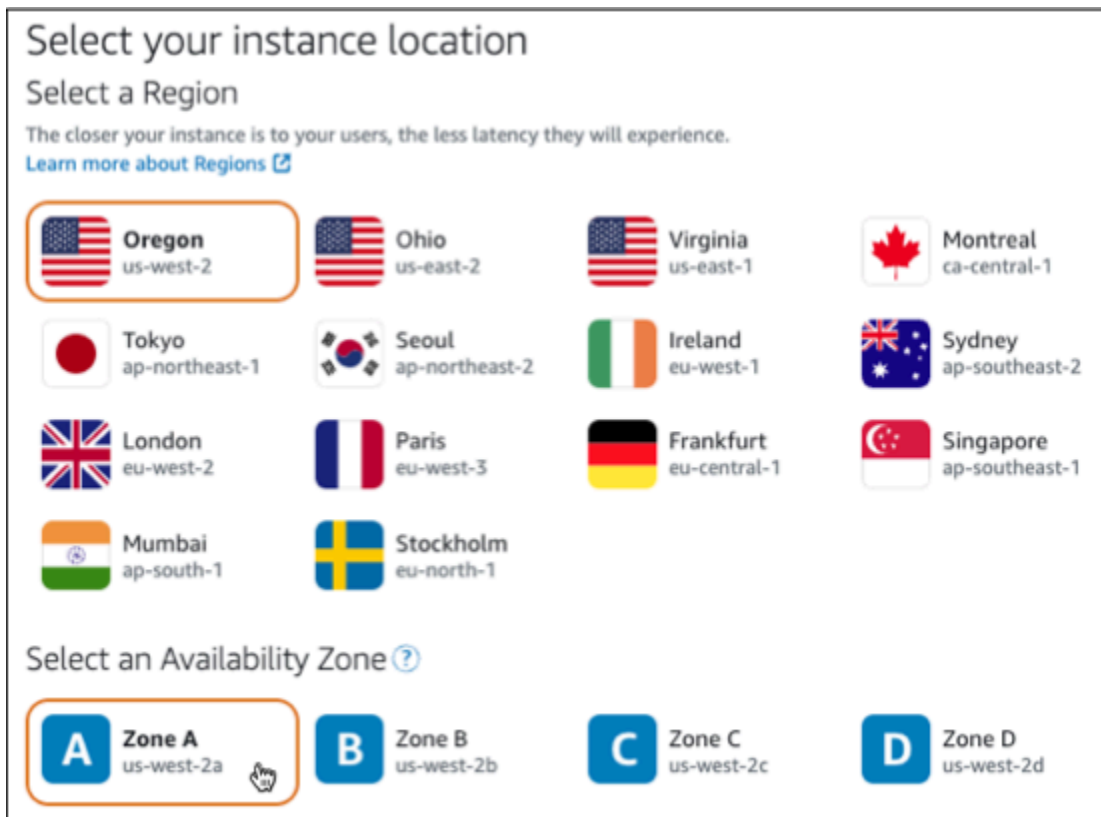
Complete the following steps to get your WordPress instance up and running.

To create a Lightsail instance for WordPress

1. Sign in to the [Lightsail console](#).
2. On the **Instances** section of the Lightsail home page, choose **Create instance**.



3. Choose the AWS Region and Availability Zone for your instance.



4. Choose the image for your instance as follows:
 - a. For **Select a platform**, choose **Linux/Unix**.
 - b. For **Select a blueprint**, choose **WordPress**.
5. Choose an instance plan.

A plan includes a machine configuration (RAM, SSD, vCPU) at a low, predictable cost, plus a data transfer allowance.

6. Enter a name for your instance. Resource names:
 - Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
7. Choose **Create instance**.
8. To view the test blog post, go to the instance management page and copy the public IPv4 address shown in the upper-right corner of the page. Paste the address into the address field of an internet-connected web browser. The browser displays the test blog post.

Step 2: Configure your WordPress instance

You can configure your WordPress instance using a guided, step-by-step workflow that configures the following:

- **A registered domain name** – Your WordPress site needs a domain name that is easy to remember. Users will specify this domain name to access your WordPress site. For more information, see [Domains and DNS](#).
- **DNS management** – You must decide how to manage the DNS records for your domain. A DNS record tells the DNS server which IP address or hostname a domain or subdomain is associated with. A DNS zone contains the DNS records for your domain. For more information, see [the section called “DNS in Lightsail”](#).
- **A Static IP address** – The default public IP address for your WordPress instance changes if you stop and start your instance. When you attach a static IP address to your instance, it stays the same even if you stop and start your instance. For more information, see [the section called “IP addresses”](#).
- **An SSL/TLS certificate** – After you create a validated certificate and install it on your instance, you can enable HTTPS for your WordPress website so that traffic that is routed to the instance through your registered domain is encrypted using HTTPS. For more information, see [the section called “Enable HTTPS”](#).

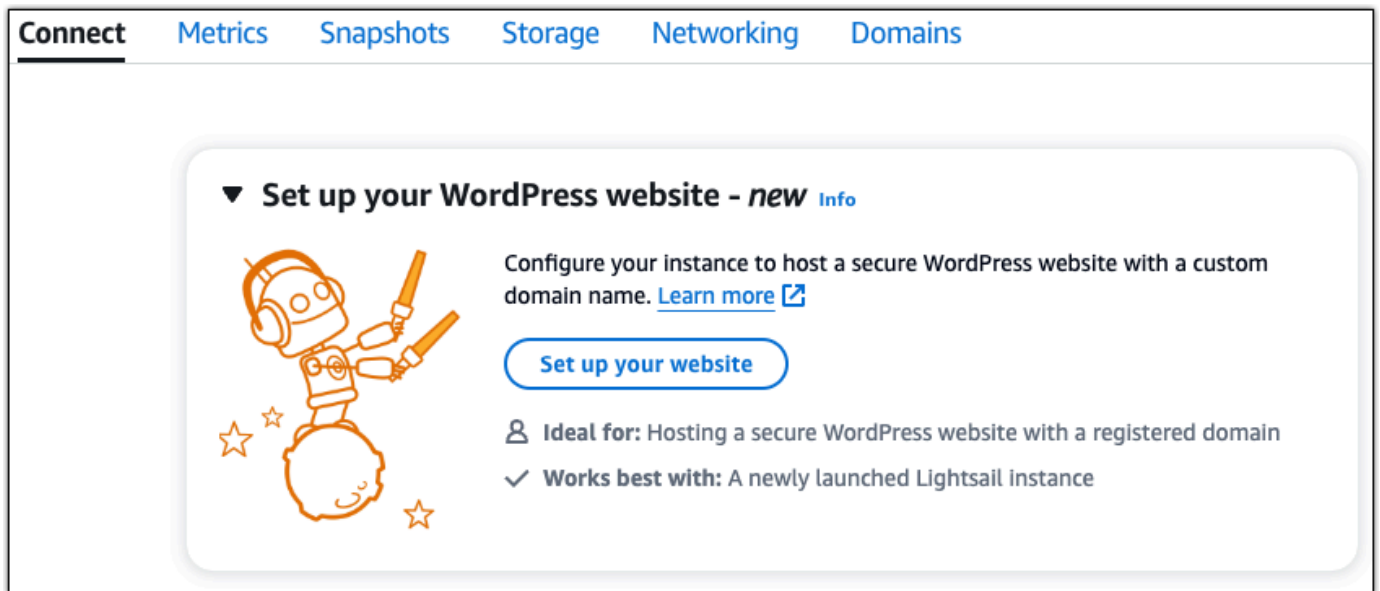
Tip

Review the following tips before you begin. For troubleshooting information, see [Troubleshooting WordPress setup](#).

- Setup supports Lightsail instances with WordPress version 6 and newer, that were created after January 1, 2023.
- Your instance must be in a **Running** state. Allow a few minutes for the SSH connection to become ready if the instance was just started.
- Ports 22, 80, and 443 on your instance firewall must allow TCP connections from any IP address while setup is running. For more information, see [Instance firewalls](#).
- When you add or update DNS records that point traffic from your apex domain (example.com) and its www subdomains (www.example.com), they will need to propagate throughout the Internet. You can verify that your DNS changes have taken effect by using tools such as [nslookup](#), or [DNS Lookup](#) from *MxToolbox*.
- Wordpress instances that were created prior to January 1, 2023, might contain a deprecated Certbot Personal Package Archive (PPA) repository that will cause website setup to fail. If this repository is present during setup, it will be removed from the existing path and backed up to the following location on your instance: `~/opt/bitnami/lightsail/repo.backup`. For more information about the deprecated PPA, see [Certbot PPA](#) on the *Canonical* website.
- Let's Encrypt certificates will automatically renew every 60 to 90 days.
- While setup is in progress, do not stop or make changes to your instance. It can take up to 15 minutes to configure your instance. You can view the progress for each step in the instance connect tab.

To configure your instance using the website setup wizard

1. On the instance management page, on the **Connect** tab, choose **Set up your website**.



The screenshot shows the Amazon Lightsail console interface. At the top, there are navigation tabs: **Connect**, **Metrics**, **Snapshots**, **Storage**, **Networking**, and **Domains**. The **Connect** tab is selected. Below the navigation, there is a large white card with a rounded border. The card has a title: **▼ Set up your WordPress website - new** with an **Info** link. To the left of the text is a cartoon illustration of a robot wearing a headset and holding a wrench and a screwdriver, with stars around it. To the right of the illustration, the text reads: "Configure your instance to host a secure WordPress website with a custom domain name. [Learn more](#)". Below this text is a blue button labeled **Set up your website**. Underneath the button, there are two lines of text: "👤 **Ideal for:** Hosting a secure WordPress website with a registered domain" and "✓ **Works best with:** A newly launched Lightsail instance".

2. For **Specify a domain name**, use an existing Lightsail managed domain, register a new domain with Lightsail, or use a domain that you registered by using another domain registrar. Choose **Use this domain** to go to the next step.
3. For **Configure DNS**, do one of the following:
 - Choose **Lightsail managed domain** to use a Lightsail DNS zone. Choose **Use this DNS zone** to go to the next step.
 - Choose **Third-party domain** to use the hosting service that manages the DNS records for your domain. Note that we create a matching DNS zone in your Lightsail account in case you decide to use it later on. Choose **Use third-party DNS** to go to the next step.
4. For **Create a static IP address**, enter a name for your static IP address and then choose **Create static IP**.
5. For **Manage domain assignments**, choose **Add assignment**, choose a domain type, and then choose **Add**. Choose **Continue** to go to the next step.
6. For **Create an SSL/TLS certificate**, choose your domains and subdomains, enter an email address, select **I authorize Lightsail to configure a Let's Encrypt certificate on my instance**, and choose **Create certificate**. We start to configure the Lightsail resources.

While setup is in progress, do not stop or make changes to your instance. It can take up to 15 minutes to configure your instance. You can view the progress for each step in the instance connect tab.

7. After the website setup is complete, verify that the URLs that you specified in the domain assignments step open your WordPress site.

Step 3: Get the default application password for your WordPress website

You need the default application password to sign in to the administration dashboard for your WordPress website.

To get the default password for the WordPress administrator

1. Open the instance management page for your WordPress instance.
2. On the **WordPress** panel, choose **Retrieve default password**. This expands **Access default password** at the bottom of the page.

The screenshot shows the AWS Lightsail console for a WordPress instance named 'WordPress-1'. The instance is running and has 1 GB RAM, 2 vCPUs, and 40 GB SSD. The 'Access WordPress Admin' section is expanded, showing the 'Default WordPress admin password' field with a 'Retrieve default password' button highlighted in a red box. Other details include the AWS Region (Virginia, Zone A), Public IPv4 address, and Instance status (Running).

3. Choose **Launch CloudShell**. This opens a panel at the bottom of the page.
4. Choose **Copy** and then paste the contents into the CloudShell window. You can either put your cursor at the CloudShell prompt and press Ctrl+V, or you can right-click to open the menu and then choose **Paste**.
5. Make a note of the password displayed in the CloudShell window. You need this to sign in to the administration dashboard of your WordPress website.

```
[cloudshell-user@ip-10-114-1-17 ~]$ AWS_REGION=us-east-1 ~/lightsail_connect WordPress-1 cat bitnami_application_password
JKzh8wB5FAR!
```

Step 4: Sign in to your WordPress website

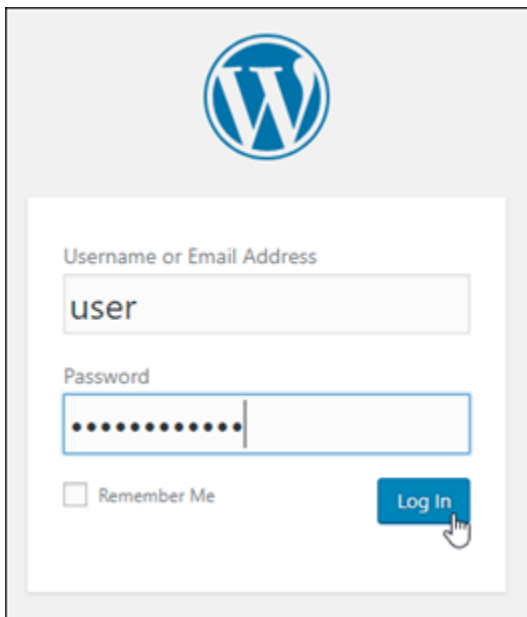
Now that you have the default user password, navigate to your WordPress website's home page, and sign in to the administration dashboard. After you're signed in, you can change the default password.

To sign in to the administration dashboard

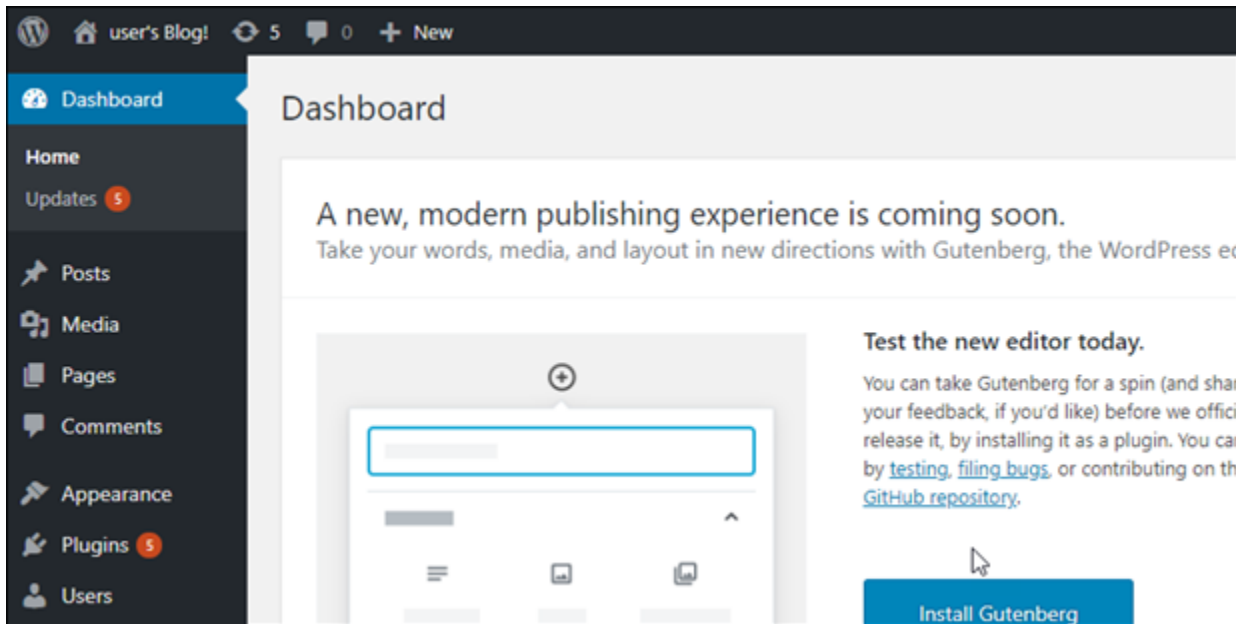
1. Open the instance management page for your WordPress instance.
2. On the **WordPress** panel, choose **Access WordPress Admin**.
3. On the **Access your WordPress Admin Dashboard** panel, under **Use public IP address**, choose the link with this format:

`http://public-ipv4-address./wp-admin`

4. For **Username or Email Address**, enter **user**.
5. For **Password**, enter the password obtained in the previous step.
6. Choose **Log in**.

A screenshot of the WordPress login interface. At the top center is the WordPress logo (a blue 'W' inside a circle). Below it is a white login form with a light gray border. The form has two input fields: the first is labeled 'Username or Email Address' and contains the text 'user'; the second is labeled 'Password' and contains a series of black dots. Below the password field is a checkbox labeled 'Remember Me' which is unchecked. To the right of the checkbox is a blue button labeled 'Log In' with a white mouse cursor icon pointing at it.

You are now signed in to the administration dashboard of your WordPress website where you can perform administrative actions. For more information about administering your WordPress website, see the [WordPress Codex](#) in the WordPress documentation.



Step 5: Read the Bitnami documentation

Read the Bitnami documentation to learn how to perform administrative tasks on your WordPress website, such as install plugins, customize the theme, and upgrade your version of WordPress.

For more information, see the [Bitnami WordPress for AWS Cloud](#).

Quick start guide: WordPress Multisite

Here are a few steps you should take to get started after your WordPress Multisite instance is up and running on Amazon Lightsail:

Contents

- [Step 1: Read the Bitnami documentation](#)
- [Step 2: Get the default application password to access the WordPress administration dashboard](#)
- [Step 3: Attach a static IP address to your instance](#)
- [Step 4: Sign in to the administration dashboard of your WordPress Multisite website](#)
- [Step 5: Route traffic for your registered domain name to your WordPress Multisite website](#)
- [Step 6: Add blogs as domains or subdomains to your WordPress Multisite website](#)
- [Step 7: Read the WordPress Multisite documentation and continue configuring your website](#)
- [Step 8: Create a snapshot of your instance](#)

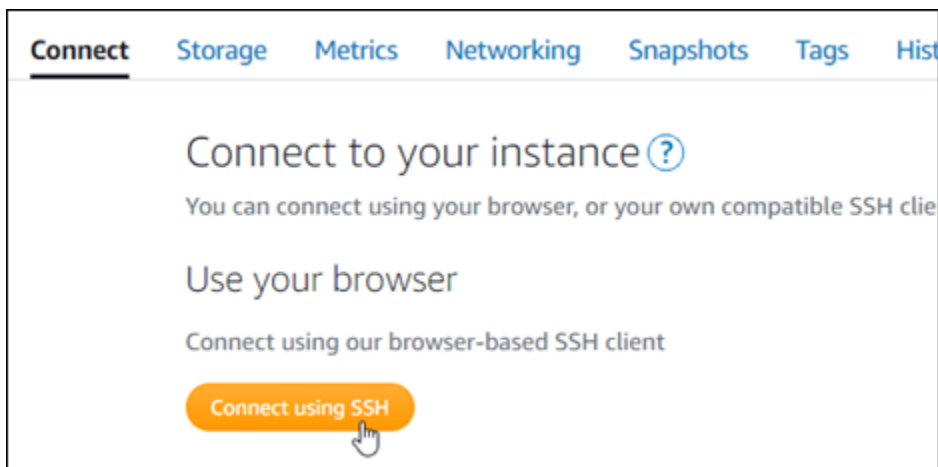
Step 1: Read the Bitnami documentation

Read the Bitnami documentation to learn how to configure your WordPress Multisite instance. For more information, see the [WordPress Multisite Packaged By Bitnami For AWS Cloud](#).

Step 2: Get the default application password to access the WordPress administration dashboard

Complete the following procedure to get the default application password required to access the administration dashboard for your WordPress Multisite website. For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

1. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command to get the default application password:

```
cat $HOME/bitnami_application_password
```

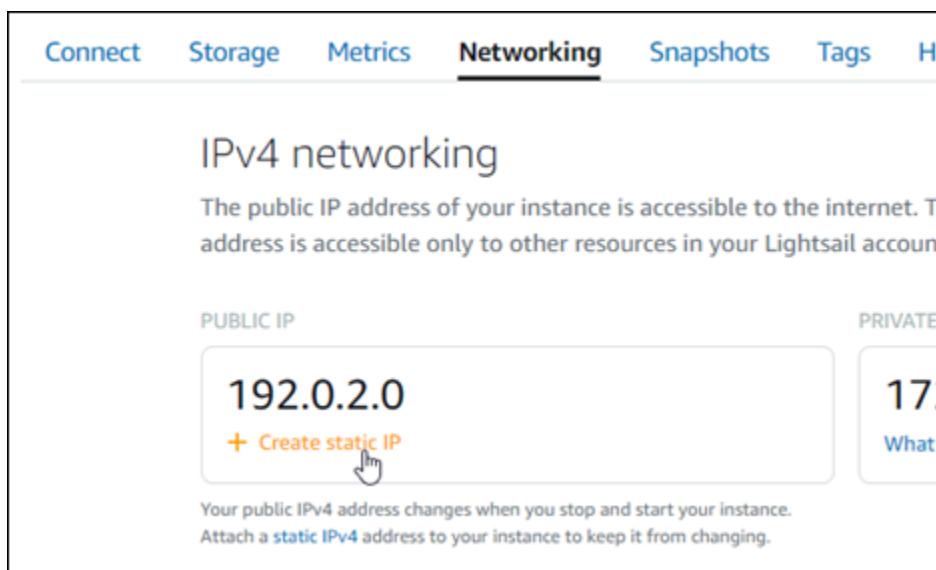
You should see a response similar to the following example, which contains the default application password. Use this password to sign in to the administration dashboard of your WordPress Multisite website.

```
bitnami@ip-172-31-28-100:~$ cat bitnami_application_password
JeVN8xDWlCIp
bitnami@ip-172-31-28-100:~$
```

Step 3: Attach a static IP address to your instance

The public IP address assigned to your instance when you first create it will change every time you stop and start your instance. You should create and attach a static IP address to your instance to ensure its public IP address doesn't change. Later, when you use your registered domain name, such as `example.com`, with your instance, you don't have to update the domain name system (DNS) of your domain every time you stop and start your instance. You can attach one static IP to an instance.

On the instance management page, under the **Networking** tab, choose **Create a static IP** or **Attach static IP** (if you previously created a static IP that you can attach to your instance), then follow the instructions on the page. For more information, see [Create a static IP and attach it to an instance](#).

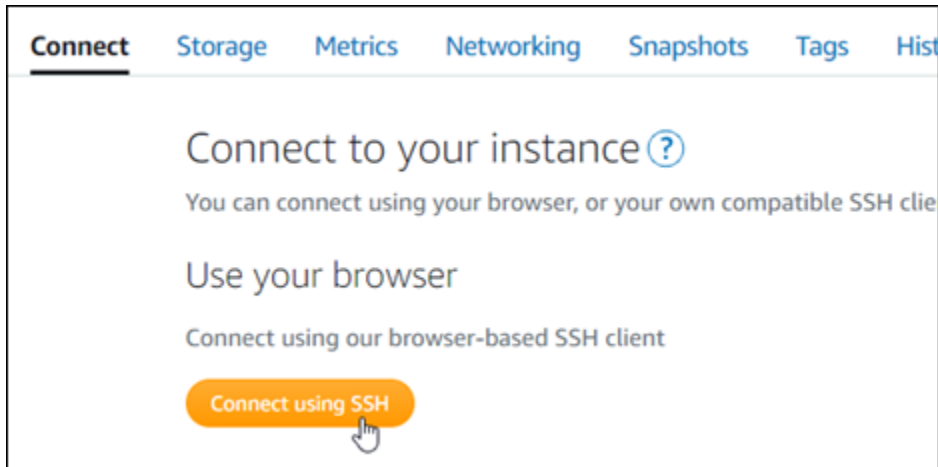


After the new static IP address is attached to your instance, you must complete the following procedure to make WordPress aware of the new static IP address.

1. Make a note of the new static IP address of your instance. It's listed in the header section of your instance management page.



2. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



3. After you're connected, enter the following command. Replace *<StaticIP>* with the new static IP address of your instance.

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

You should see a response similar to the following example. The WordPress website on your instance should now be aware of the new static IP address.

```
bitnami@ip-173-36-0-197:~$ sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
Configuring domain to 203.0.113.0
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO ==> Updating hostname in database
prestashop 15:49:22.46 INFO ==> Purging cache
Disabling automatic domain update for IP address changes
```

If that command fails, you might be using an older version of the WordPress Multisite instance. Try running the following commands instead. Replace *<StaticIP>* with the new static IP address of your instance.

```
cd /opt/bitnami/apps/wordpress
sudo ./bnconfig --machine_hostname <StaticIP>
```

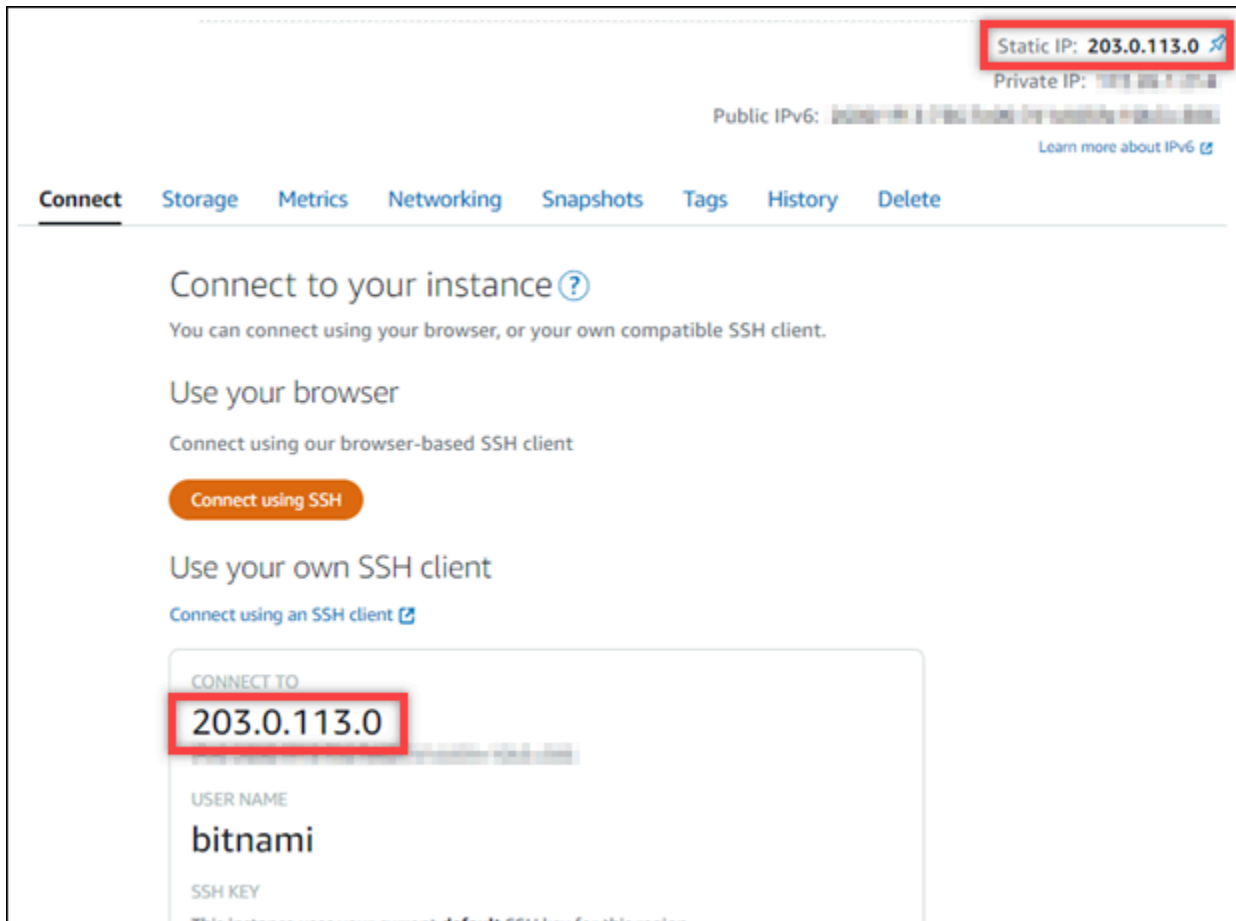
After running those commands, enter the following command to keep the bnconfig tool from automatically running every time the server restarts.

```
sudo mv bnconfig bnconfig.disabled
```

Step 4: Sign in to the administration dashboard of your WordPress Multisite website

Now that you have the default application password, complete the following procedure to navigate to your WordPress Multisite website's home page, and sign in to the administration dashboard. After you're signed in, you can start customizing your website and making administrative changes. For more information about what you can do in WordPress, see the [Step 7: Read the WordPress Multisite documentation and continue configuring your website](#) section later in this guide.

1. On your instance management page, under the **Connect** tab, make note of the public IP address of your instance. The public IP address is also displayed in the header section of your instance management page.



2. Browse to the public IP address of your instance, for example by going to `http://203.0.113.0`.

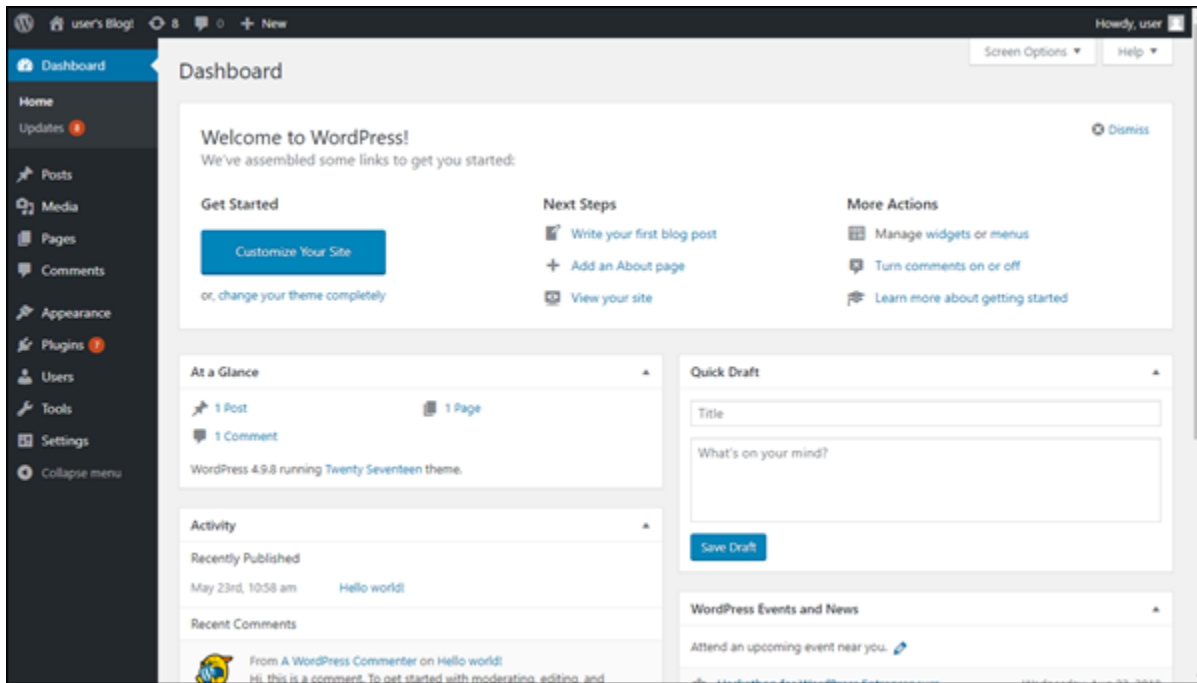
The home page of your WordPress website should appear.

3. Choose **Manage** in the bottom right corner of your WordPress website home page.

If the **Manage** banner is not shown, you can reach the sign in page by browsing to `http://<PublicIP>/wp-login.php`. Replace `<PublicIP>` with the public IP address of your instance.

4. Sign in using the default user name (`user1`) and the default password retrieved earlier in this guide.

The WordPress administration dashboard appears.



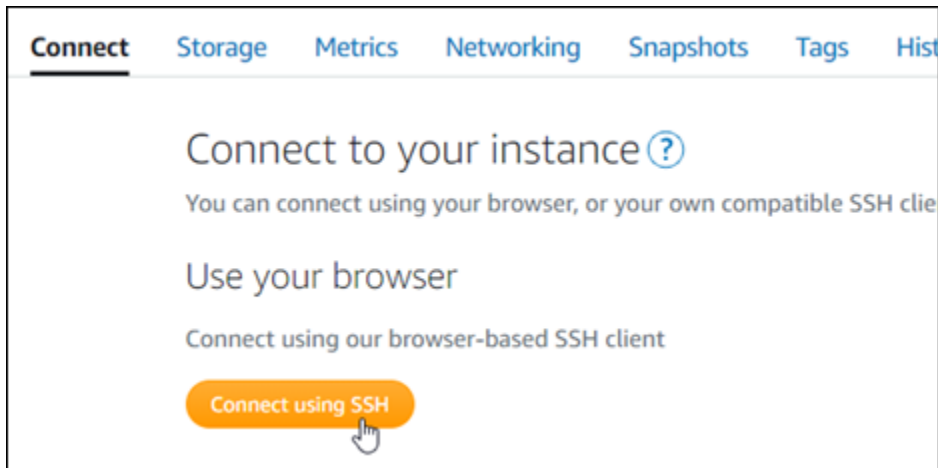
Step 5: Route traffic for your registered domain name to your WordPress Multisite website

To route traffic for your registered domain name, such as `example.com`, to your WordPress Multisite website, you add a record to the DNS of your domain. DNS records are typically managed and hosted at the registrar where you registered your domain. However, we recommend that you transfer management of your domain's DNS records to Lightsail so that you can administer it using the Lightsail console.

On the Lightsail console home page, under the **Domains & DNS** tab, choose **Create DNS zone**, then follow the instructions on the page. For more information, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

After your domain name is routing traffic to your instance, you must complete the following procedure to make WordPress aware of the domain name.

1. On the instance management page, under the **Connect** tab, choose **Connect using SSH**.



2. After you're connected, enter the following command. Replace *<DomainName>* with the domain name that is routing traffic to your instance.

```
sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain www.example.com
```

You should see a response similar to the following example. The WordPress Multisite software should now be aware of the domain name.

```
bitnami@ip-173-20-0-199:~$ sudo /opt/bitnami/configure_app_domain --domain www.example.com
Configuring domain to www.example.com
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO ==> Updating hostname in database
prestashop 15:49:22.46 INFO ==> Purging cache
Disabling automatic domain update for IP address changes
```

If that command fails, you might be using an older version of the WordPress Multisite instance. Try running the following commands instead. Replace *<DomainName>* with the domain name that is routing traffic to your instance.

```
cd /opt/bitnami/apps/wordpress
sudo ./bnconfig --machine_hostname <DomainName>
```

After running those commands, enter the following command to keep the bnconfig tool from automatically running every time the server restarts.

```
sudo mv bnconfig bnconfig.disabled
```

If you browse to the domain name that you configured for your instance, you should be redirected to the main blog of your WordPress Multisite website. Next you must decide whether you want to add blogs as domains or as subdomains to your WordPress Multisite website. For more information, continue to the next [Step 6: Add blogs as domains or subdomains to your WordPress Multisite website](#) section of this guide.

Step 6: Add blogs as domains or subdomains to your WordPress Multisite website

WordPress Multisite is designed to host multiple blog websites on one instance of WordPress. When you add new blog websites to your WordPress Multisite, you can configure them to use their own domains or a subdomain of your WordPress Multisite's primary domain. You can configure your WordPress Multisite to use only one of those options. For example, if you choose to add blog sites as domains, then you cannot add blog sites as subdomains, and vice versa. To configure either of those options, see one of the following guides:

- To add blog sites as domains, such as `example1.com` and `example2.com`, see [Add blogs as domains to your WordPress Multisite instance in Lightsail](#).
- To add blog sites as subdomains of your WordPress Multisite's primary domain, such as `one.example.com` and `two.example.com`, see [Add blogs as subdomains to your WordPress Multisite instance in Lightsail](#).

Step 7: Read the WordPress Multisite documentation and continue configuring your website

Read the WordPress Multisite documentation to learn how to administer and customize your website. For more information, see the [WordPress Multisite Network Administration Documentation](#).

Step 8: Create a snapshot of your instance

After you configure your WordPress Multisite website the way you want it, create periodic snapshots of your instance to back it up. You can create snapshots manually, or enable automatic snapshots to have Lightsail create daily snapshots for you. If something goes wrong with your

instance, you can create a new replacement instance using the snapshot. For more information, see [Snapshots](#).

On the instance management page, under the **Snapshot** tab, choose **Create a snapshot** or choose to enable automatic snapshots.

The screenshot shows the 'Snapshots' tab in the Amazon Lightsail console. At the top, there are navigation tabs: Connect, Storage, Metrics, Networking, **Snapshots**, Tags, History, and Delete. The 'Snapshots' tab is selected and underlined.

Manual snapshots (with a help icon):

You can create a snapshot to back up your instance, its system disk, and attached disks.

+ Create snapshot

>	<input type="checkbox"/> February 5, 2021 - 9:37 AM	"Prestashop-1612546662"	⋮
>	<input type="checkbox"/> January 13, 2021 - 9:44 AM	"Prestashop-1610559880"	⋮
>	<input type="checkbox"/> December 9, 2020 - 12:33 PM	"Prestashop-1607545986"	⋮
>	<input type="checkbox"/> September 9, 2020 - 5:44 PM	"Prestashop-1599698658"	⋮

Showing 4 of 4 snapshots

Automatic snapshots (with a help icon):

Automatic snapshots are enabled

Your daily snapshot time is 10:00 PM PST.
We will store your seven most recent snapshots.

Change snapshot time

DAILY SNAPSHOTS

>	<input type="checkbox"/> Thursday	March 4, 2021	⋮
>	<input type="checkbox"/> Wednesday	March 3, 2021	⋮
>	<input type="checkbox"/> Tuesday	March 2, 2021	⋮

For more information, see [Creating a snapshot of your Linux or Unix instance in Amazon Lightsail](#) or [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

Bitnami tutorials for Amazon Lightsail

Bitnami simplifies software application deployment by providing pre-packaged and ready-to-run development stacks and applications for various platforms. Use the following tutorials to learn how to work with Bitnami in Lightsail.

Topics

- [Get the application user name and password for your Bitnami instance](#)
- [Remove the Bitnami banner from a Bitnami blueprint instance in Lightsail](#)

Get the application user name and password for your Bitnami instance

Bitnami provides many of the application instance images, or blueprints, that you can create as Amazon Lightsail instances, which are your virtual private servers. These blueprints are described as “Packaged by Bitnami” in the instance creation page in the Lightsail console.

After you create an instance using a Bitnami blueprint, you sign in and administer it. To do this, you must get the default user name and password for the application and/or database running on the instance. This article shows you how to obtain the information necessary to sign in and administer Lightsail instances created from the following blueprints:

- WordPress blogging and content management application
- WordPress Multisite blogging and content management application with support for multiple websites on the same instance
- Django development stack
- Ghost blogging and content management application
- LAMP development stack (PHP 7)
- Node.js development stack
- Joomla content management application
- Magento e-Commerce application
- MEAN development stack
- Drupal content management application
- GitLab CE repository application
- Redmine project management application

- Nginx (LEMP) development stack

Get the default Bitnami application and database user name

These are the default application and database user names for Lightsail instances created using the Bitnami blueprints:

Note

Not all Bitnami blueprints include an application or a database. The user name is listed as not applicable (N/A) when these are not included in the blueprint.

- WordPress, including WordPress Multisite
 - Application user name: `user`
 - Database user name: `root`
- PrestaShop
 - Application user name: `user@example.com`
 - Database user name: `root`
- Django
 - Application user name: N/A
 - Database user name: `root`
- Ghost
 - Application user name: `user@example.com`
 - Database user name: `root`
- LAMP stack (PHP 5 and PHP 7)
 - Application user name: N/A
 - Database user name: `root`
- Node.js
 - Application user name: N/A
 - Database user name: N/A
- Joomla
 - Application user name: `user`

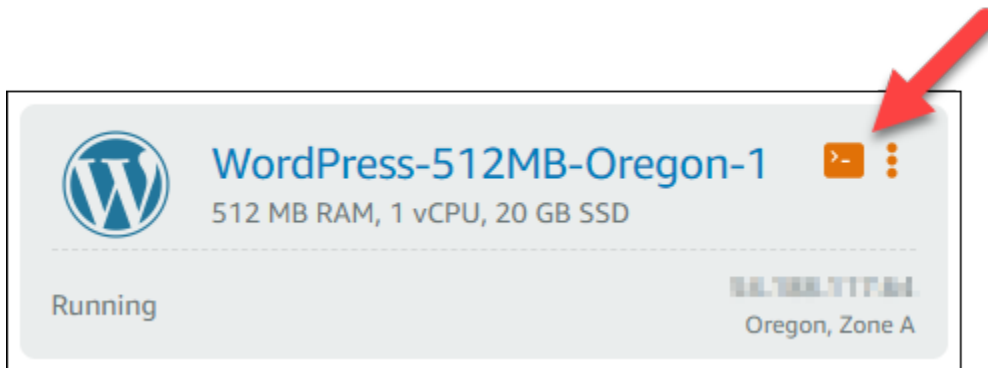
- Database user name: root
- Magento
 - Application user name: user
 - Database user name: root
- MEAN
 - Application user name: N/A
 - Database user name: root
- Drupal
 - Application user name: user
 - Database user name: root
- GitLab CE
 - Application user name: user
 - Database user name: postgres
- Redmine
 - Application user name: user
 - Database user name: root
- Nginx
 - Application user name: N/A
 - Database user name: root

Get the default Bitnami application and database password

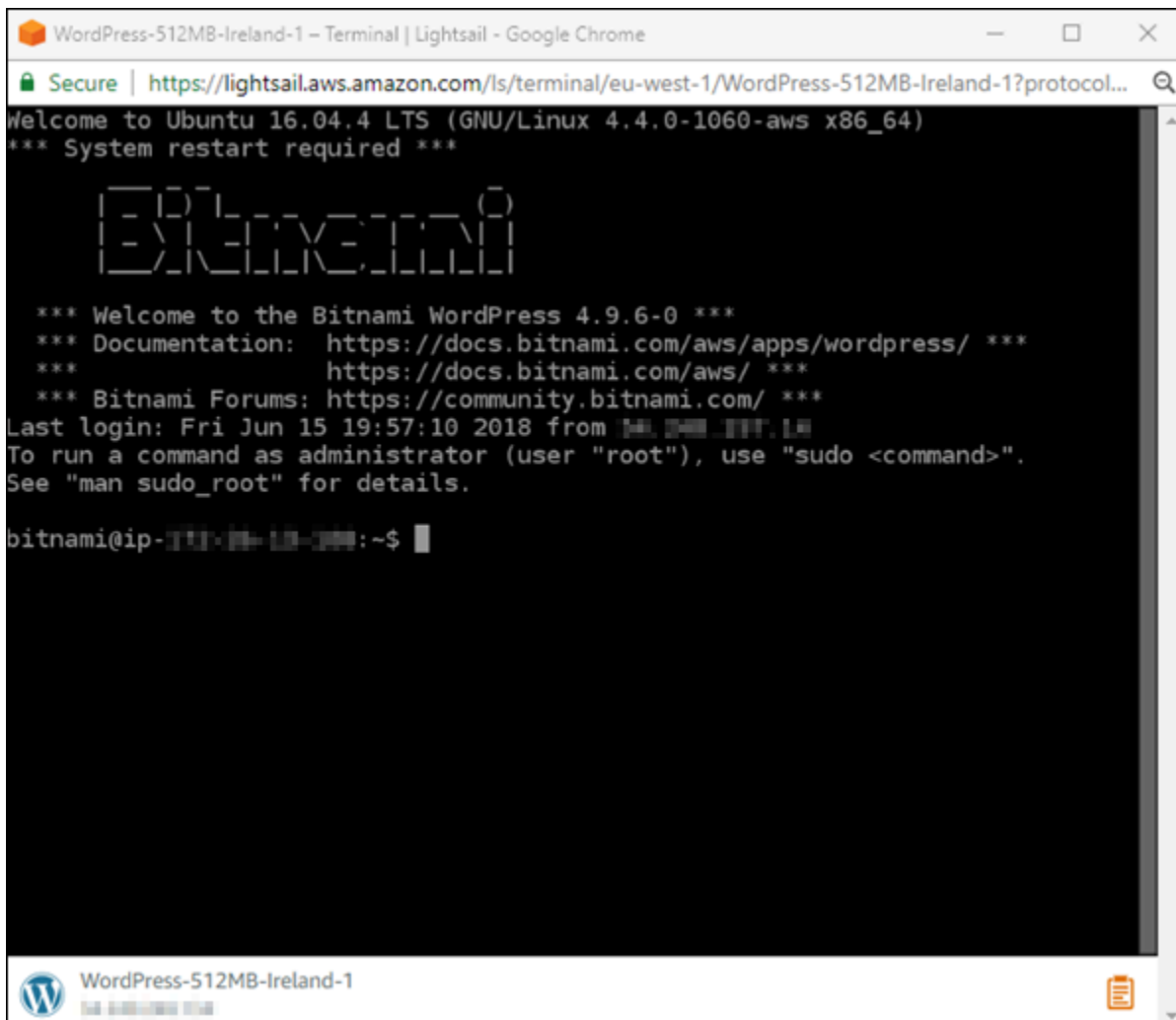
The default application and database password are stored on your instance. You retrieve it by connecting to it using the browser-based SSH terminal in the Lightsail console and running a special command.

To get the default Bitnami application and database password

1. Sign in to the [Lightsail console](#).
2. If you haven't already, create an instance using a Bitnami blueprint. For more information, see [Create an Amazon Lightsail VPS](#)
3. On the Lightsail home page, choose the quick connect icon for the instance you want to connect to.



The browser-based SSH client window opens, as shown in the following example.



4. Type the following command to retrieve the default application password:

```
cat bitnami_application_password
```

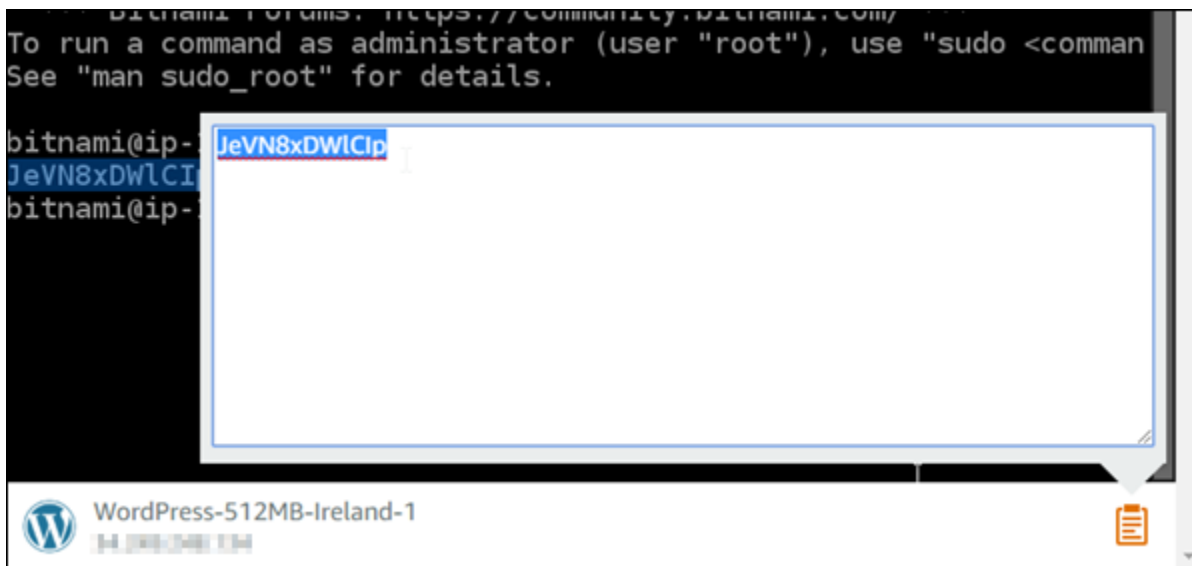
Note

If you're in a directory other than the user home directory, then type `cat $HOME/bitnami_application_password`.

You should see a response similar to this, which contains the application password:

```
bitnami@ip-172-31-33-100:~$ cat bitnami_application_password
JeVN8xDwLCIp
bitnami@ip-172-31-33-100:~$
```

5. In the terminal screen, highlight the password, then choose the clipboard icon in the bottom right corner of the browser-based SSH client window.
6. In the clipboard text box, highlight the text you want to copy, then press **Ctrl+C** or **Cmd+C** to copy the text to your local clipboard.

**Important**

Make sure to save your password somewhere at this time. You can change it later after you sign in to the Bitnami application on your instance.

Sign in to the Bitnami application on your instance

For instances created from the WordPress, Joomla, Magento, Drupal, GitLab CE, and Redmine blueprints, sign in to the application by browsing to the public IP address of your instance.

To sign in to the Bitnami application

1. In a browser window, navigate to the public IP address for your instance.

The Bitnami application home page opens. The home page displays according to the Bitnami blueprint you chose for your instance. For example, this is the WordPress application home page:

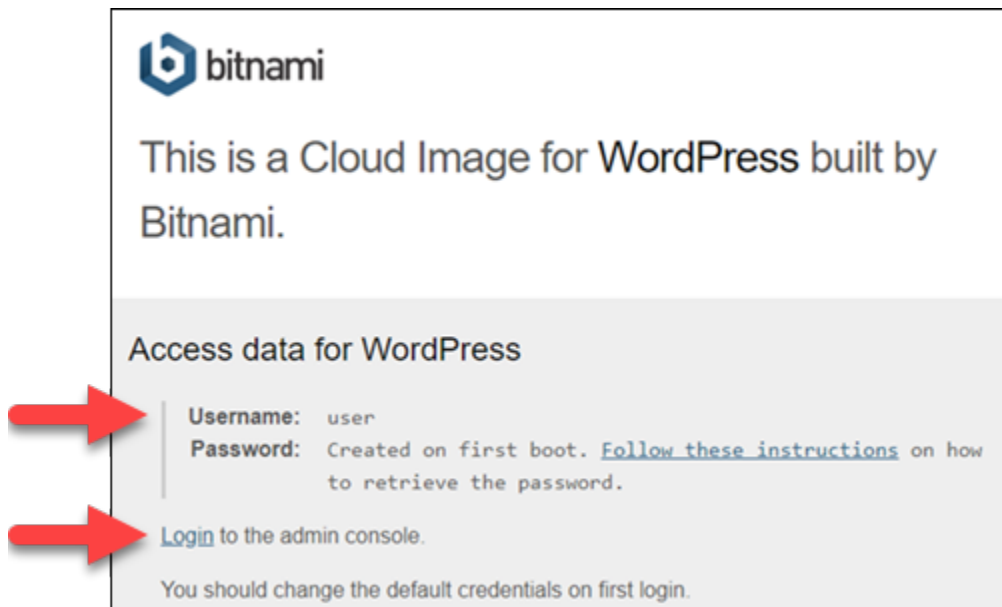


2. Choose the Bitnami logo at the bottom right corner of the application home page to go to the application information page.

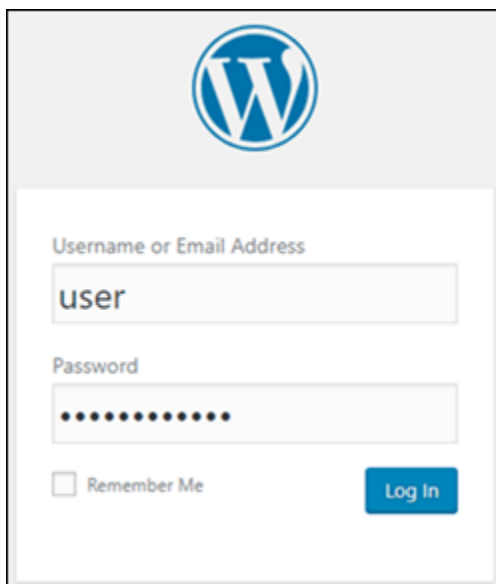
Note

The GitLab CE application doesn't display a Bitnami logo. Instead, sign in using the user name and password text fields displayed on the GitLab CE home page.

The application information page contains the default user name and a link to the login page for the application on your instance.



3. Choose the login link on the page to go to the log in page for the application on your instance.
4. Type the user name and the password you just acquired, then choose **Log In**.



Next steps

Use the following links to learn more about the Bitnami blueprints and view their tutorials. For example, you can [install plugins](#) or [enable HTTPS support with SSL certificates](#) for your WordPress instance.

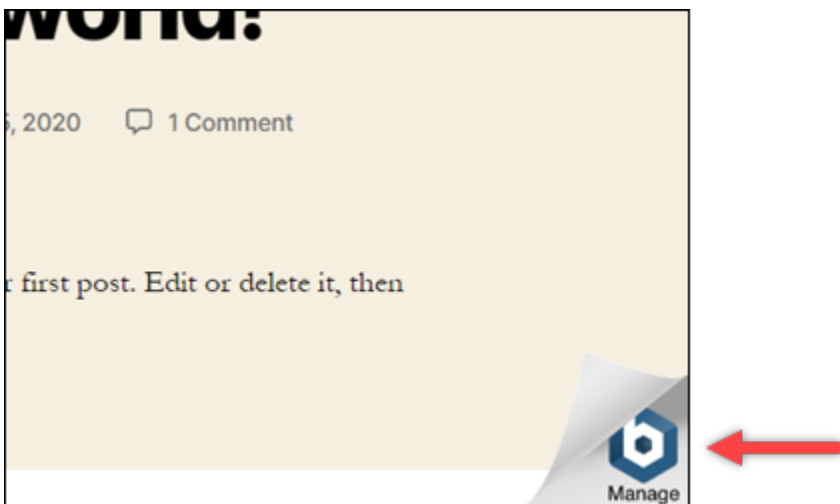
- [Bitnami WordPress for Amazon Web Services](#)
- [Bitnami LAMP stack for Amazon Web Services](#)

- [Bitnami Node.js for Amazon Web Services](#)
- [Bitnami Joomla for Amazon Web Services](#)
- [Bitnami Magento for Amazon Web Services](#)
- [Bitnami MEAN stack for Amazon Web Services](#)
- [Bitnami Drupal for Amazon Web Services](#)
- [Bitnami GitLab for Amazon Web Services](#)
- [Bitnami Redmine for Amazon Web Services](#)
- [Bitnami Nginx \(LEMP stack\) for Amazon Web Services](#)

For more information, see [Get Started with Bitnami Applications using Amazon Lightsail](#) or [Using Amazon Lightsail FAQ](#).

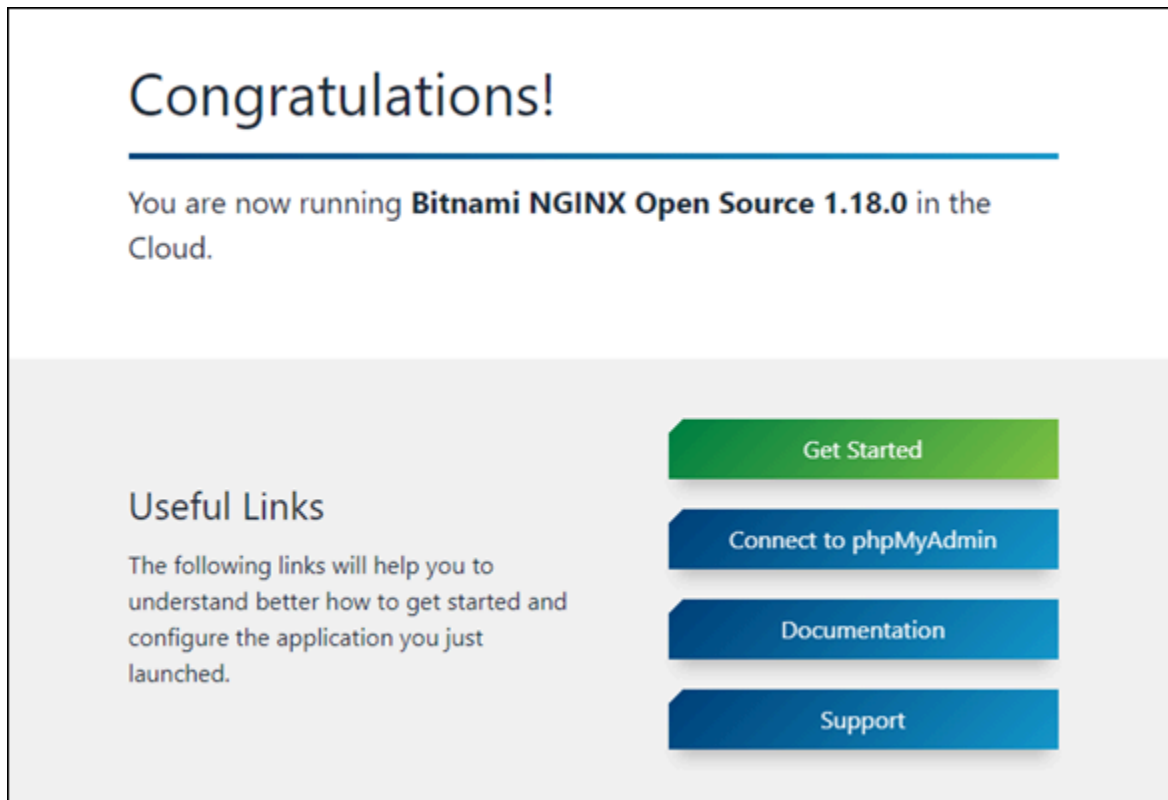
Remove the Bitnami banner from a Bitnami blueprint instance in Lightsail

Some of the Bitnami blueprints that can be selected for Amazon Lightsail instances display a Bitnami banner on the home page of the application. In the following example from a "Certified by Bitnami" WordPress instance, the Bitnami banner is displayed in the bottom-right corner of the home page. In this guide, we show you how to permanently remove the Bitnami icon from the home page of the application on your instance.



Not all Bitnami blueprint applications display the Bitnami banner on the home page of the application. Visit the home page of your Lightsail instance to determine if a Bitnami banner is displayed. In the following example from a "Packaged by Bitnami" Nginx instance, the Bitnami icon

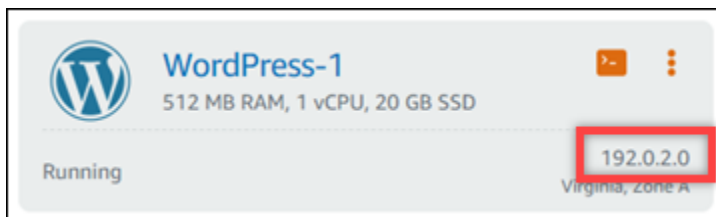
is not displayed. Instead, a place-holder information page is displayed, which is eventually replaced by the application that you choose to deploy on your instance. If your instance doesn't display a Bitnami banner, then you don't have to follow the procedures in this guide.



Remove the Bitnami banner from your instance

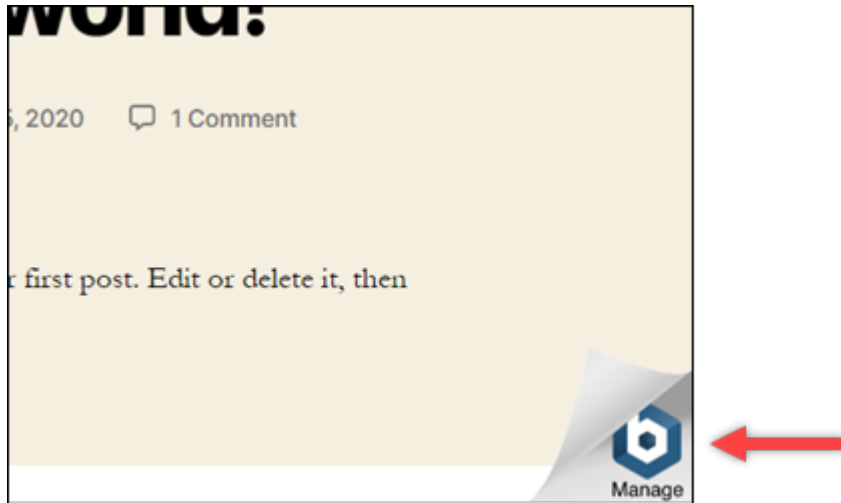
Complete the following procedure to confirm that your instance has a Bitnami icon displayed in the home page of the application, and to remove it.

1. Sign in to the [Lightsail console](#).
2. In the **Instances** tab of the Lightsail home page, copy the public IP address of the instance that you want to confirm.



3. Open a new browser tab, enter the public IP address of your instance into the address bar, and press **Enter**.
4. Confirm one of the following options:

1. If the Bitnami icon is not displayed on the page, then stop following these procedures. You don't need to remove the Bitnami icon from the home page of your application.
2. If the Bitnami icon is displayed in the lower-right corner of the page as shown in the following example, then continue to the following set of steps to remove it.



In the following set of steps, you will connect to your instance using the Lightsail browser-based SSH client. After you're connected, you will run the Bitnami Configuration Tool (bnconfig) tool to remove the Bitnami icon from the home page of your application. The bnconfig tool is a command line tool that allows you to configure your application on your Bitnami blueprint instance. For more information, see [Learn About The Bitnami Configuration Tool](#) in the *Bitnami documentation*.

5. Return to the browser tab that is on the Lightsail home page.
6. Choose the browser-based SSH client icon that is next to the name of the instance that you wish to connect to.



7. After the SSH client is connected to your instance, enter one of the following commands:
 1. If your instance uses Apache, then enter one of the following commands. If one of the commands fails, try the other. The first part of this command disables the Bitnami banner, and the second part restarts the Apache service.

```
sudo /opt/bitnami/apps/wordpress/bnconfig --disable_banner 1 && sudo /opt/bitnami/ctlscript.sh restart apache
```

```
sudo /opt/bitnami/wordpress/bnconfig --disable_banner 1 && sudo /opt/bitnami/ctlscript.sh restart apache
```

You can confirm that the process was successful by browsing to the public IP address of your instance and confirming that the Bitnami icon is gone.

WordPress tutorials for Amazon Lightsail

WordPress is an open-source content management system that allows users to create and manage websites and blogs with ease. Use the following tutorials to learn how to work with WordPress in Lightsail.

Tasks

- [Tutorial: Launch and configure a WordPress instance in Lightsail](#)
- [Tutorial: Connect a WordPress website in Lightsail to an Amazon S3 bucket](#)
- [Tutorial: Connect a WordPress instance in Lightsail to an Amazon Aurora database](#)
- [Tutorial: Connect a WordPress website to a MySQL managed database in Lightsail](#)
- [Tutorial: Connect a WordPress instance to a Lightsail bucket](#)
- [Configure your WordPress instance to work with a content delivery network distribution in Lightsail](#)
- [Enable email on your WordPress instance in Lightsail](#)
- [Enable HTTPS on your WordPress instance in Lightsail](#)
- [Migrate an existing WordPress blog to Amazon Lightsail](#)

Tutorial: Launch and configure a WordPress instance in Lightsail

Amazon Lightsail is the easiest way to get started with Amazon Web Services (AWS) if you just need instances (virtual private servers). Lightsail includes everything you need to launch your project quickly—instances, managed databases, SSD-based storage, backups (snapshots), data transfer, domain DNS management, static IPs, and load balancers—for a [low, predictable price](#).

With this tutorial, you'll learn how to launch and configure a WordPress instance on Lightsail. It includes steps to configure a custom domain name, secure internet traffic with HTTPS, connect to your instance by using SSH, and sign in to your WordPress website. When you're done with this tutorial, you have the fundamentals to get your instance up and running on Lightsail.

Note

As part of the AWS Free Tier, you can get started with Amazon Lightsail for free on select instance bundles. For more information, see **AWS Free Tier** on the [Amazon Lightsail Pricing page](#).

Contents

- [Step 1: Sign up for AWS](#)
- [Step 2: Create a WordPress instance](#)
- [Step 3: Configure your WordPress instance](#)
- [Step 4: Get the admin password for your WordPress website](#)
- [Step 5: Sign in to the administration dashboard of your WordPress website](#)
- [Additional information](#)

Step 1: Sign up for AWS

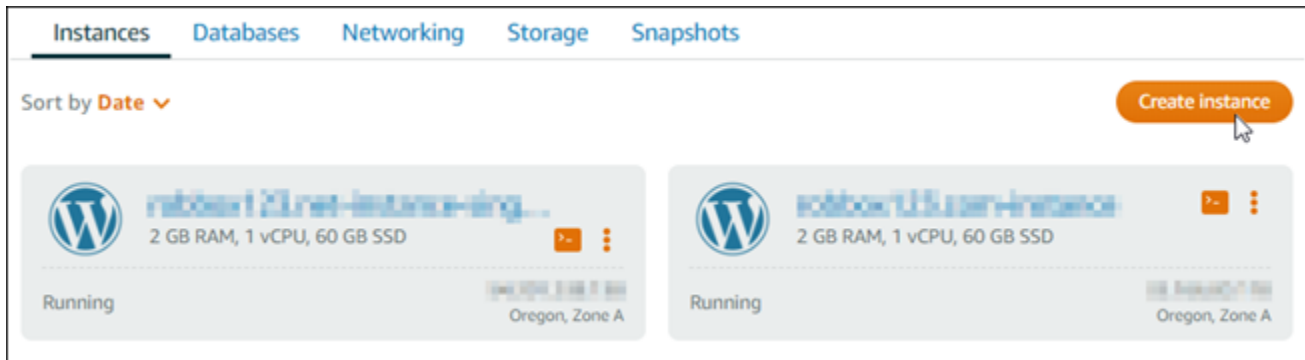
Amazon Lightsail requires an AWS account. [Sign up for AWS](#), or [sign in to AWS](#) if you already have an account.

Step 2: Create a WordPress instance

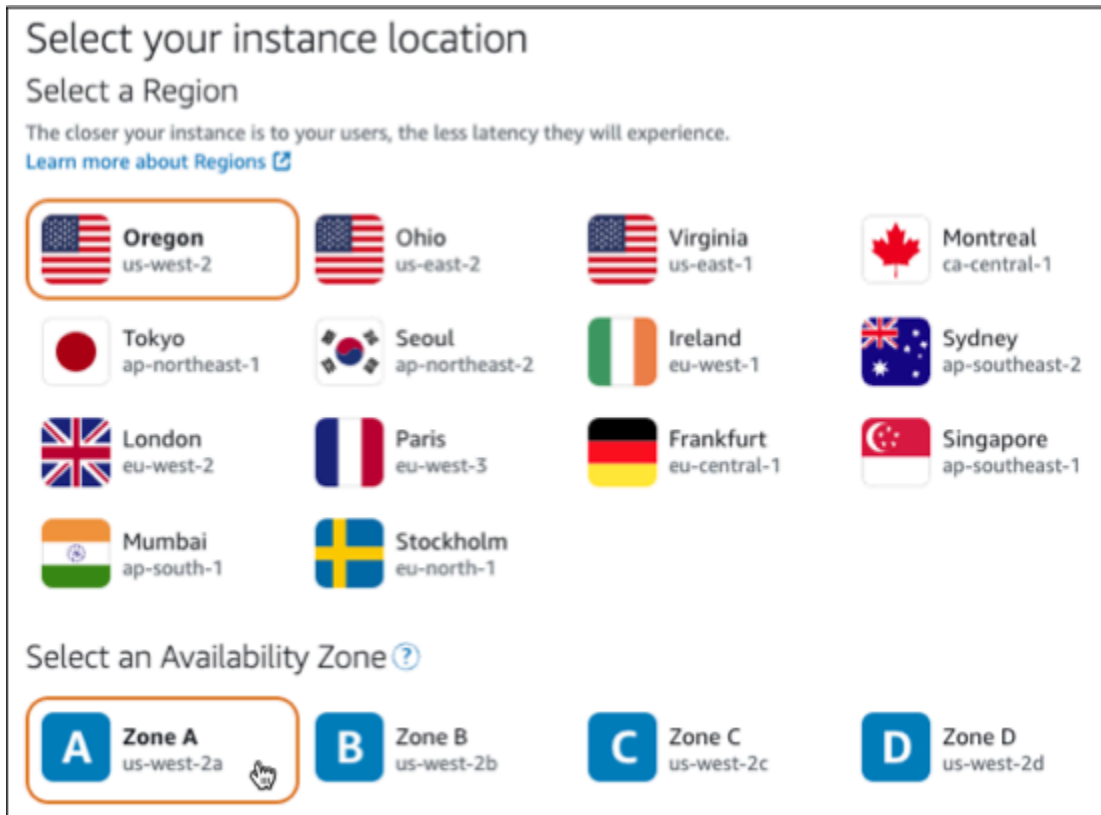
Complete the following steps to get your WordPress instance up and running. For more information, see [the section called "Create an instance"](#).

To create a Lightsail instance for WordPress

1. Sign in to the [Lightsail console](#).
2. On the **Instances** section of the Lightsail home page, choose **Create instance**.



3. Choose the AWS Region and Availability Zone for your instance.



4. Choose the image for your instance as follows:
 - a. For **Select a platform**, choose **Linux/Unix**.
 - b. For **Select a blueprint**, choose **WordPress**.
5. Choose an instance plan.

A plan includes a machine configuration (RAM, SSD, vCPU) at a low, predictable cost, plus a data transfer allowance.

6. Enter a name for your instance. Resource names:

- Must be unique within each AWS Region in your Lightsail account.
 - Must contain 2 to 255 characters.
 - Must start and end with an alphanumeric character or number.
 - Can include alphanumeric characters, numbers, periods, dashes, and underscores.
7. Choose **Create instance**.
 8. To view the test blog post, go to the instance management page and copy the public IPv4 address shown in the upper-right corner of the page. Paste the address into the address field of an internet-connected web browser. The browser displays the test blog post.

Step 3: Configure your WordPress instance

You can configure your WordPress instance by using a guided, step-by-step workflow, or you can complete the individual tasks. Using either option, you will configure the following:

- **A registered domain name** – Your WordPress site needs a domain name that is easy to remember. Users will specify this domain name to access your WordPress site. For more information, see [Domains and DNS](#).
- **DNS management** – You must decide how to manage the DNS records for your domain. A DNS record tells the DNS server which IP address or hostname a domain or subdomain is associated with. A DNS zone contains the DNS records for your domain. For more information, see [the section called “DNS in Lightsail”](#).
- **A Static IP address** – The default public IP address for your WordPress instance changes if you stop and start your instance. When you attach a static IP address to your instance, it stays the same even if you stop and start your instance. For more information, see [the section called “IP addresses”](#).
- **An SSL/TLS certificate** – After you create a validated certificate and install it on your instance, you can enable HTTPS for your WordPress website so that traffic that is routed to the instance through your registered domain is encrypted using HTTPS. For more information, see [the section called “Enable HTTPS”](#).

Option: Guided workflow

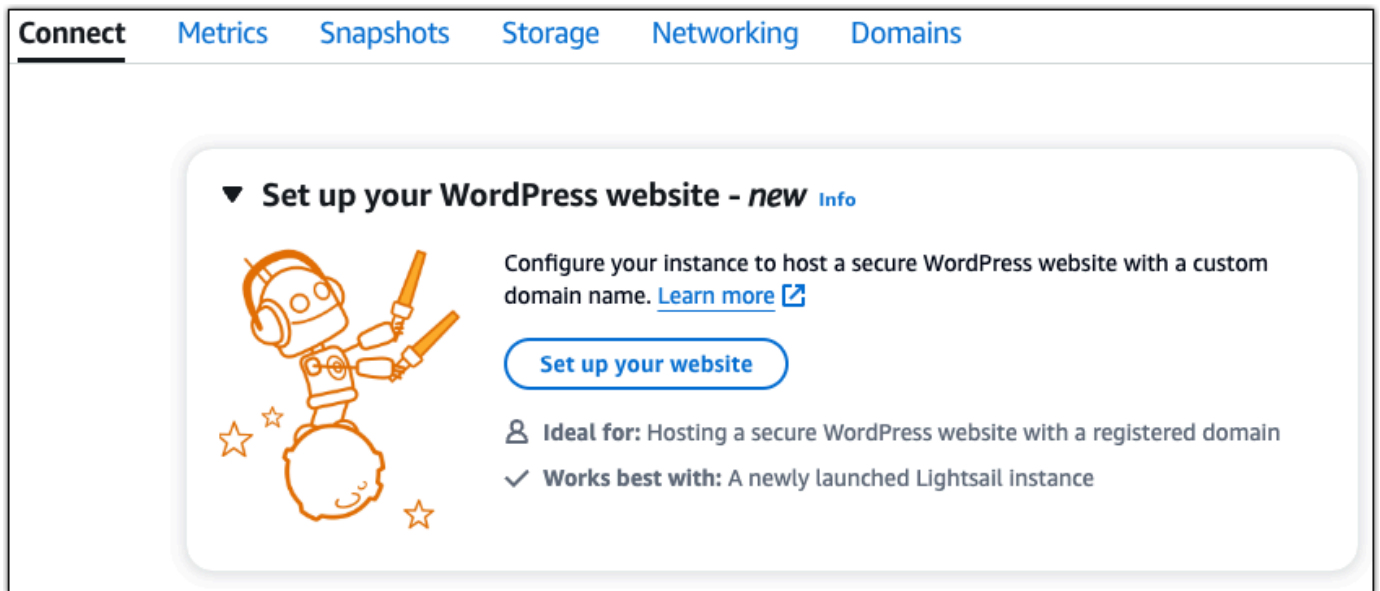
Tip

Review the following tips before you begin. For troubleshooting information, see [Troubleshooting WordPress setup](#).

- Setup supports Lightsail instances with WordPress version 6 and newer, that were created after January 1, 2023.
- Your instance must be in a **Running** state. Allow a few minutes for the SSH connection to become ready if the instance was just started.
- Ports 22, 80, and 443 on your instance firewall must allow TCP connections from any IP address while setup is running. For more information, see [Instance firewalls](#).
- When you add or update DNS records that point traffic from your apex domain (example.com) and its www subdomains (www.example.com), they will need to propagate throughout the Internet. You can verify that your DNS changes have taken effect by using tools such as [nslookup](#), or [DNS Lookup](#) from *MxToolbox*.
- Wordpress instances that were created prior to January 1, 2023, might contain a deprecated Certbot Personal Package Archive (PPA) repository that will cause website setup to fail. If this repository is present during setup, it will be removed from the existing path and backed up to the following location on your instance: `~/opt/bitnami/lightsail/repo.backup`. For more information about the deprecated PPA, see [Certbot PPA](#) on the *Canonical* website.
- Let's Encrypt certificates will automatically renew every 60 to 90 days.
- While setup is in progress, do not stop or make changes to your instance. It can take up to 15 minutes to configure your instance. You can view the progress for each step in the instance connect tab.

To configure your instance using the website setup wizard

1. On the instance management page, on the **Connect** tab, choose **Set up your website**.



The screenshot shows the Amazon Lightsail console interface. At the top, there are navigation tabs: **Connect**, **Metrics**, **Snapshots**, **Storage**, **Networking**, and **Domains**. The **Connect** tab is selected. Below the navigation, there is a card titled "Set up your WordPress website - *new* [Info](#)". The card features an illustration of a robot wearing headphones and holding tools, surrounded by stars. To the right of the illustration, the text reads: "Configure your instance to host a secure WordPress website with a custom domain name. [Learn more](#)". Below this text is a blue button labeled "Set up your website". Underneath the button, there are two bullet points: "Ideal for: Hosting a secure WordPress website with a registered domain" and "Works best with: A newly launched Lightsail instance".

2. For **Specify a domain name**, use an existing Lightsail managed domain, register a new domain with Lightsail, or use a domain that you registered by using another domain registrar. Choose **Use this domain** to go to the next step.
3. For **Configure DNS**, do one of the following:
 - Choose **Lightsail managed domain** to use a Lightsail DNS zone. Choose **Use this DNS zone** to go to the next step.
 - Choose **Third-party domain** to use the hosting service that manages the DNS records for your domain. Note that we create a matching DNS zone in your Lightsail account in case you decide to use it later on. Choose **Use third-party DNS** to go to the next step.
4. For **Create a static IP address**, enter a name for your static IP address and then choose **Create static IP**.
5. For **Manage domain assignments**, choose **Add assignment**, choose a domain type, and then choose **Add**. Choose **Continue** to go to the next step.
6. For **Create an SSL/TLS certificate**, choose your domains and subdomains, enter an email address, select **I authorize Lightsail to configure a Let's Encrypt certificate on my instance**, and choose **Create certificate**. We start to configure the Lightsail resources.

While setup is in progress, do not stop or make changes to your instance. It can take up to 15 minutes to configure your instance. You can view the progress for each step in the instance connect tab.

7. After the website setup is complete, verify that the URLs that you specified in the domain assignments step open your WordPress site.

Option: Individual tasks

To configure your instance by completing the individual tasks

1. Create a static IP address

On the instance management page, on the **Networking** tab, choose **Create static IP**. The static IP location and instance are selected for you. Specify a name for your static IP address and then choose **Create and attach**.

2. Create a DNS zone

In the navigation pane, choose **Domains & DNS**. Choose **Create DNS zone**, enter your domain, and then choose **Create DNS zone**. If web traffic is currently being routed to your domain, make sure that all of the existing DNS records are present in the Lightsail DNS zone before changing the name servers at your domain's current DNS hosting provider. This way, traffic continually flows uninterrupted after the transfer to the Lightsail DNS zone

3. Manage domain assignments

On the page for the DNS zone, on the **Assignments** tab, choose **Add assignment**. Choose the domain or subdomain, select your instance, attach the static IP address, and then choose **Assign**.

Tip

Allow time for these changes to propagate to the internet before your domain starts routing traffic to your WordPress instance.

4. Create and install an SSL/TLS certificate

For step-by-step directions, see [the section called "Enable HTTPS"](#).

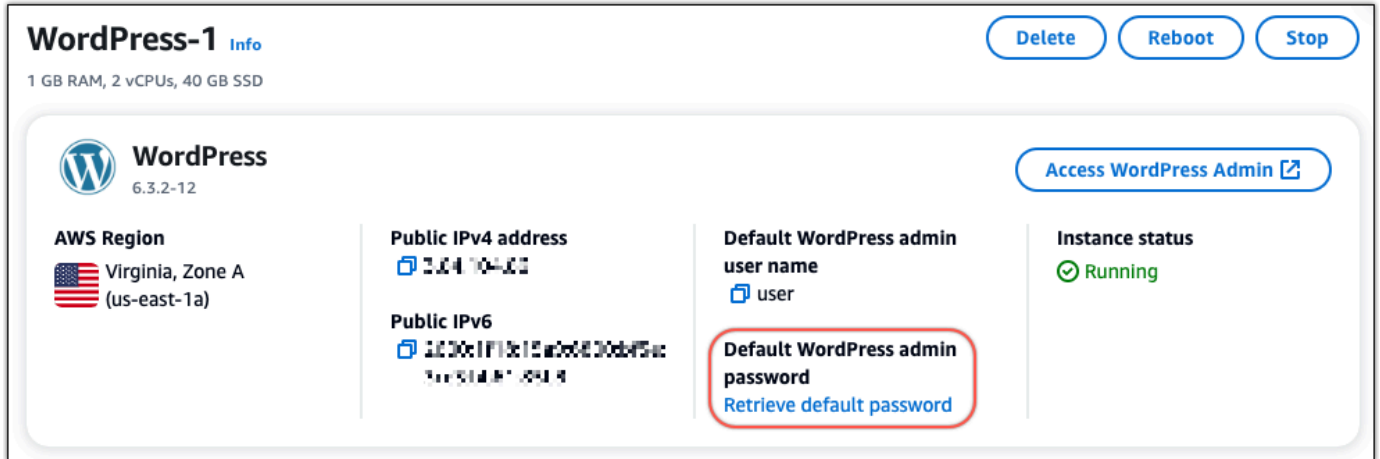
5. Verify that the URLs that you specified in the domain assignments step open your WordPress site.

Step 4: Get the admin password for your WordPress website

The default password to sign in to the administration dashboard of your WordPress website is stored on the instance. Complete the following steps to get the password.

To get the default password for the WordPress administrator

1. Open the instance management page for your WordPress instance.
2. On the **WordPress** panel, choose **Retrieve default password**. This expands **Access default password** at the bottom of the page.



3. Choose **Launch CloudShell**. This opens a panel at the bottom of the page.
4. Choose **Copy** and then paste the contents into the CloudShell window. You can either put your cursor at the CloudShell prompt and press Ctrl+V, or you can right-click to open the menu and then choose **Paste**.
5. Make a note of the password displayed in the CloudShell window. You need this to sign in to the administration dashboard of your WordPress website.

```
[cloudshell-user@ip-3-24-104-22 ~]$ AWS_REGION=us-east-1 ~/lightsail_connect WordPress-1 cat bitnami_application_password
JKzh8wB5FAR!
```

Step 5: Sign in to the administration dashboard of your WordPress website

Now that you have the password for the administration dashboard of your WordPress website, you can sign in. In the administration dashboard, you can change your user password, install plugins, change the theme of your website, and more.

Complete the following steps to sign in to the administration dashboard of your WordPress website.

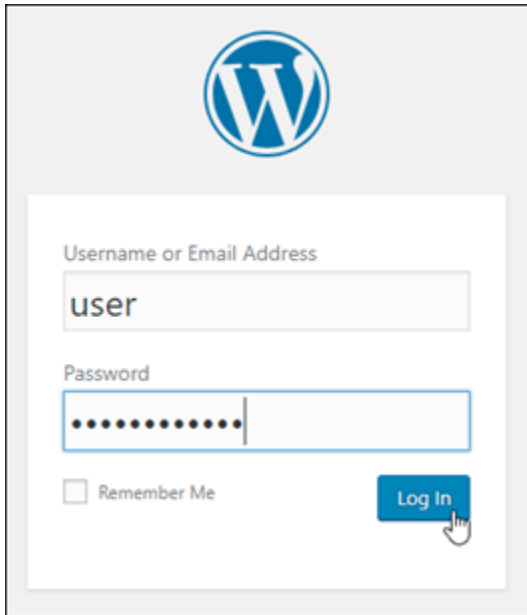
To sign in to the administration dashboard

1. Open the instance management page for your WordPress instance.

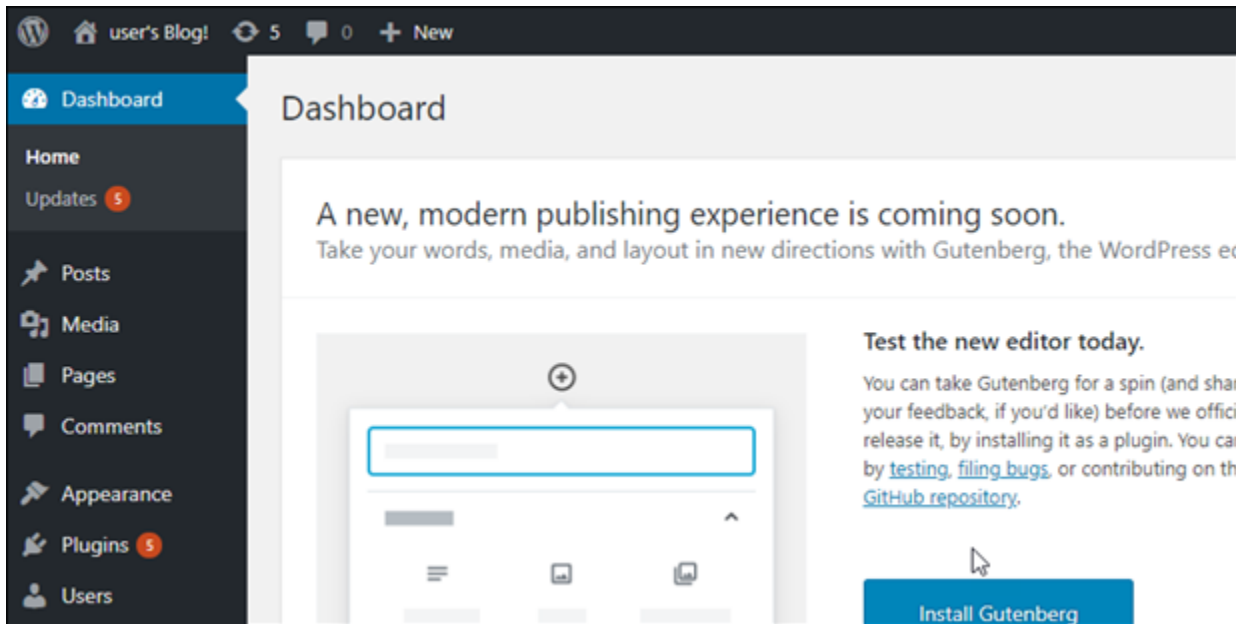
2. On the **WordPress** panel, choose **Access WordPress Admin**.
3. On the **Access your WordPress Admin Dashboard** panel, under **Use public IP address**, choose the link with this format:

`http://public-ipv4-address./wp-admin`

4. For **Username or Email Address**, enter **user**.
5. For **Password**, enter the password obtained in the previous step.
6. Choose **Log in**.

A screenshot of the WordPress login interface. At the top center is the WordPress logo, a blue 'W' inside a circle. Below the logo is a white rectangular form with a light gray border. Inside the form, there are two input fields: the first is labeled 'Username or Email Address' and contains the text 'user'; the second is labeled 'Password' and contains a series of black dots. Below the password field is a checkbox labeled 'Remember Me' which is unchecked. To the right of the checkbox is a blue button with the text 'Log In' in white. A mouse cursor is pointing at the 'Log In' button.

You are now signed in to the administration dashboard of your WordPress website where you can perform administrative actions. For more information about administering your WordPress website, see the [WordPress Codex](#) in the WordPress documentation.



Additional information

Here are some additional steps that you can perform after launching a WordPress instance in Amazon Lightsail:

- [the section called “Configure a CDN”](#)
- [Create a snapshot of your Linux or Unix instance](#)
- [Enable or disable automatic snapshots for instances or disks](#)
- [Create and attach additional block storage disks to your Linux-based instances](#)

Tutorial: Connect a WordPress website in Lightsail to an Amazon S3 bucket

This tutorial describes the steps required to connect your WordPress website running on an Amazon Lightsail instance to an Amazon Simple Storage Service (Amazon S3) bucket to store website images and attachments. To do this, you configure a WordPress plugin with a set of Amazon Web Services (AWS) account credentials. The plugin then creates the Amazon S3 bucket for you and configures your website to use the bucket instead of the instance's disk for website images and attachments.

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Install the WP Offload Media plugin on your WordPress website](#)
- [Step 3: Create an IAM user and policy](#)
- [Step 4: Edit the WordPress configuration file](#)
- [Step 5: Create the Amazon S3 bucket using the WP Offload Media plugin](#)
- [Step 6: Next steps](#)

Step 1: Complete the prerequisites

Before you get started, create a WordPress instance in Lightsail, and make sure it's in a running state. For more information, see [Tutorial: Launch and configure a WordPress instance](#).

Step 2: Install the WP Offload Media plugin on your WordPress website

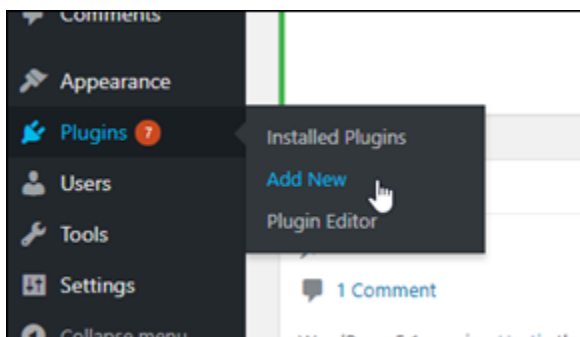
You must use a plugin to configure your website to use an Amazon S3 bucket. Many plugins are available to configure this; one such plugin is [WP Offload Media Lite](#).

Complete the following steps to install the WP Offload Media plugin on your WordPress website:

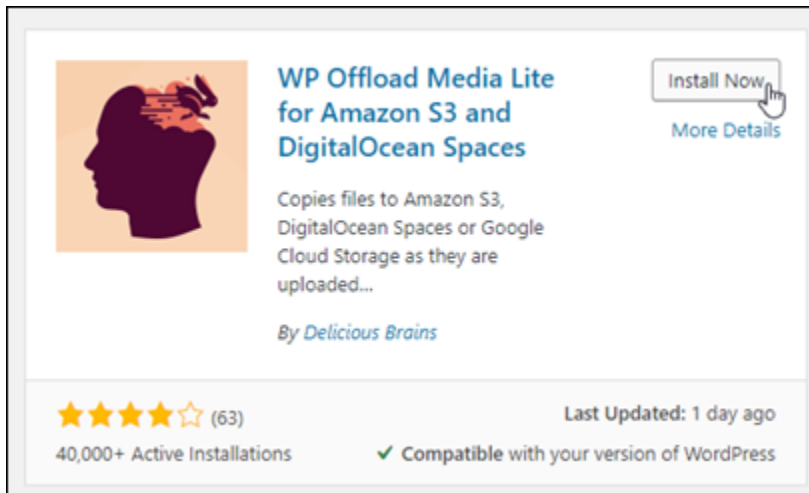
1. Sign in to your WordPress dashboard as an administrator.

For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

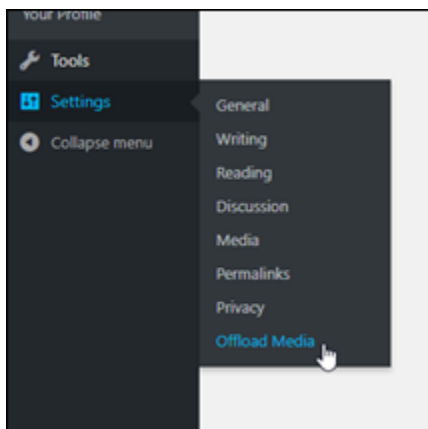
2. Hover over **Plugins** in the left navigation menu, and choose **Add New**.



3. Search for **WP Offload Media Lite**.
4. In the search results, choose **Install Now** next to the **WP Offload Media** plugin.

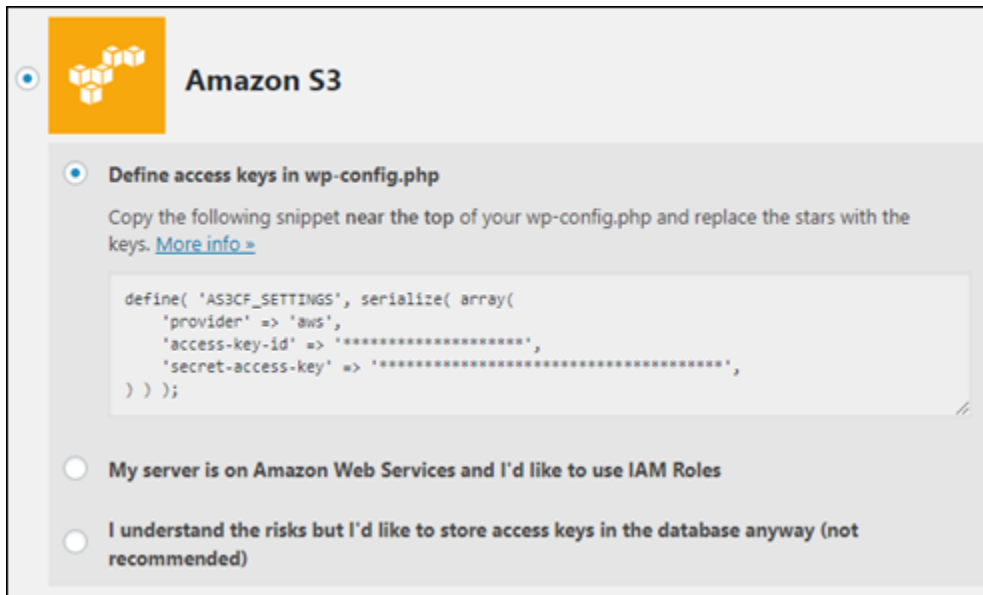


5. Choose **Activate** after the plugin is done installing.
6. In the left navigation menu, choose **Settings**, then choose **Offload Media**.



7. In the **Offload Media** page, choose **Amazon S3** as the storage provider, then choose **Define access keys in wp-config.php**.

With this option, you must add your AWS account credentials to the `wp-config.php` on the instance. These steps are covered later in this tutorial.



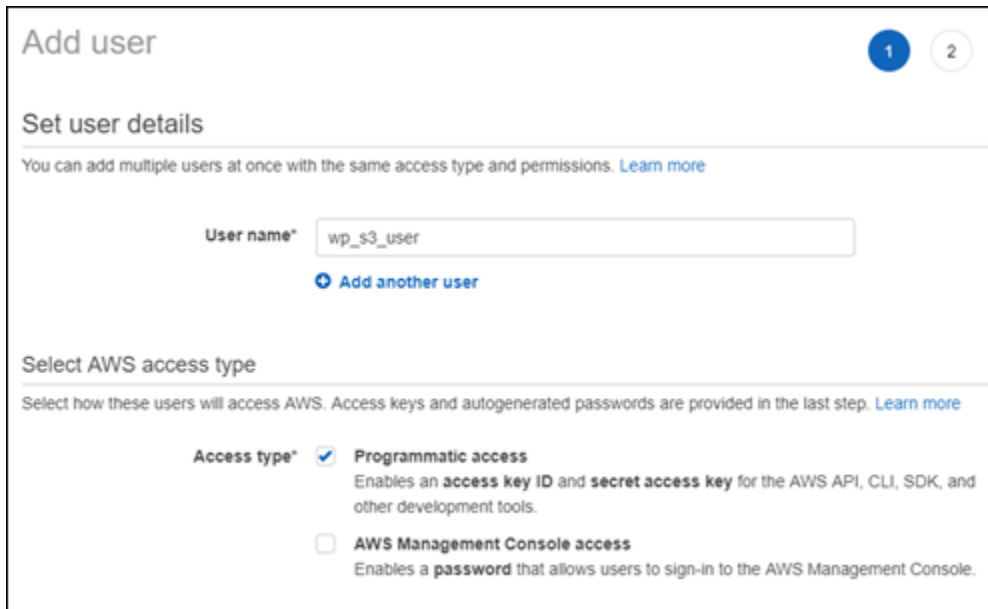
Leave the **Offload Media** page open; you will return to it later in this tutorial. Continue to the [Step 3: Create an IAM user and policy](#) section of this tutorial.

Step 3: Create an IAM user and policy

The WP Offload Media plugin requires access to your AWS account to create the Amazon S3 bucket, and to upload your website images and attachments.

Complete the following steps to create a new AWS Identity and Access Management (IAM) user and policy for the WP Offload Media plugin:

1. Open a new browser tab, and sign in to the [IAM console](#).
2. In the left navigation menu, choose **Users**.
3. Choose **Add user**.
4. In the **User name** text box, enter a name for the new user. Enter something descriptive, such as `wp_s3_user` or `wp_offload_media_plugin_user`, so that you can easily identify it in the future when performing maintenance.
5. Under the **Access type** section, choose **Programmatic access**.



Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[Add another user](#)

Select AWS access type

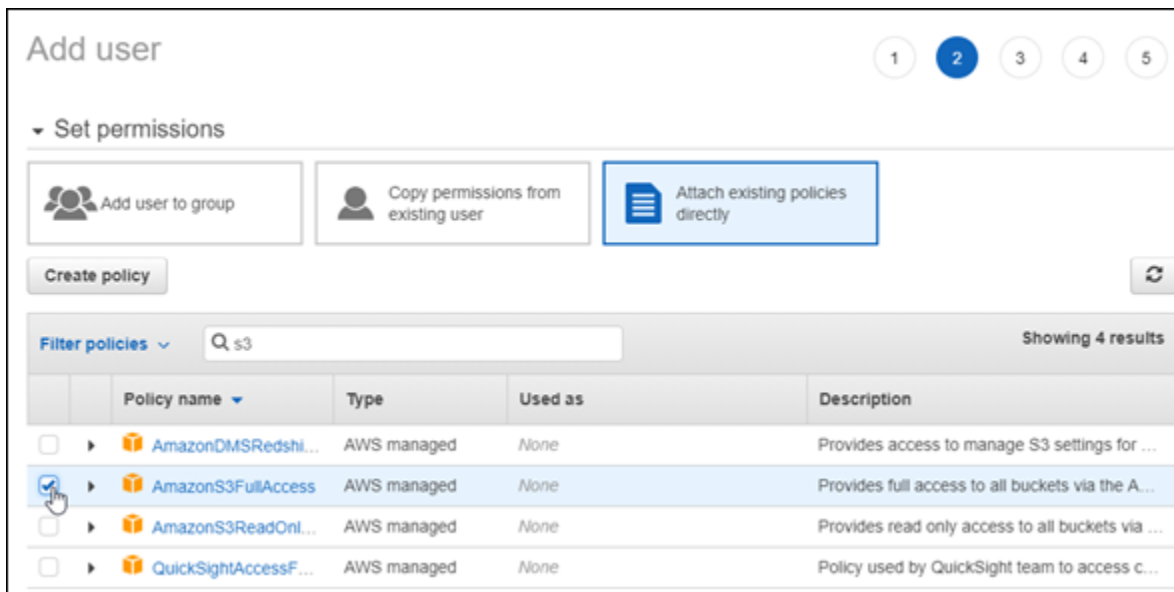
Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

6. Choose **Next: Permissions**.

7. Choose **Attach existing policies directly**, search for **S3**, and then choose **AmazonS3FullAccess** in the search results.



Add user

Set permissions

[Add user to group](#) [Copy permissions from existing user](#) [Attach existing policies directly](#)

[Create policy](#)

Filter policies Showing 4 results

	Policy name	Type	Used as	Description
<input type="checkbox"/>	AmazonDMSRedshi...	AWS managed	None	Provides access to manage S3 settings for ...
<input checked="" type="checkbox"/>	AmazonS3FullAccess	AWS managed	None	Provides full access to all buckets via the A...
<input type="checkbox"/>	AmazonS3ReadOnl...	AWS managed	None	Provides read only access to all buckets via ...
<input type="checkbox"/>	QuickSightAccessF...	AWS managed	None	Policy used by QuickSight team to access c...

8. Choose **Next: Tags**, and then choose **Next: Review**.

9. Review the user details displayed on the page, then choose **Create user**.

10. Make note of the **access key ID** and **secret access key** for the user, or choose **Download .csv** to save a copy of these values to your local drive. You will need these in the next few steps when editing the `wp-config.php` file on the WordPress instance.

Step 4: Edit the WordPress configuration file

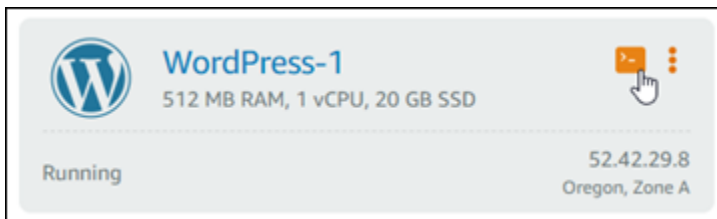
Complete the following steps to connect to your WordPress instance using the browser-based SSH client in the Lightsail console and edit the `wp-config.php` file.

The `wp-config.php` file contains your website's base configuration details, such as database connection information.

Note

You can also connect to your instance using your own SSH client. For more information, see [Download and set up PuTTY to connect using SSH in Amazon Lightsail](#)

1. Sign in to the [Lightsail console](#).
2. Choose the browser-based SSH client icon for the WordPress instance.



3. In the SSH client window that appears, enter the following command to create a backup of the `wp-config.php` file in case something goes wrong:

```
sudo cp /opt/bitnami/wordpress/wp-config.php /opt/bitnami/wordpress/wp-config.php.backup
```

4. Enter the following command to open the `wp-config.php` file using nano, a text editor:

```
nano /opt/bitnami/wordpress/wp-config.php
```

5. Enter the following text above the `/* That's all, stop editing! Happy blogging. */` text.

Be sure to replace *AccessKeyID* with the access key ID and *SecretAccessKey* with the secret access key of the IAM user you created earlier in these steps.

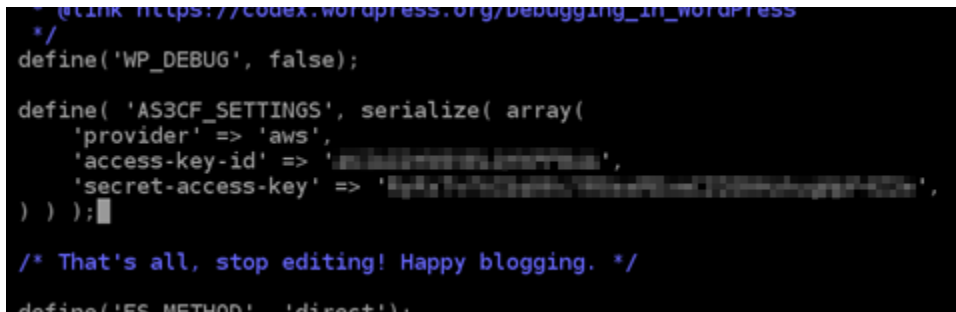
```
define( 'AS3CF_SETTINGS', serialize( array(
    'provider' => 'aws',
```

```
'access-key-id' => 'AccessKeyID',
'secret-access-key' => 'SecretAccessKey',
) ) );
```

Example:

```
define( 'AS3CF_SETTINGS', serialize( array(
    'provider' => 'aws',
    'access-key-id' => 'AKIAIOSFODNN7EXAMPLE',
    'secret-access-key' => 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
) ) );
```

The result should look like the following example:



```
define('WP_DEBUG', false);

define( 'AS3CF_SETTINGS', serialize( array(
    'provider' => 'aws',
    'access-key-id' => 'AKIAIOSFODNN7EXAMPLE',
    'secret-access-key' => 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
) ) );

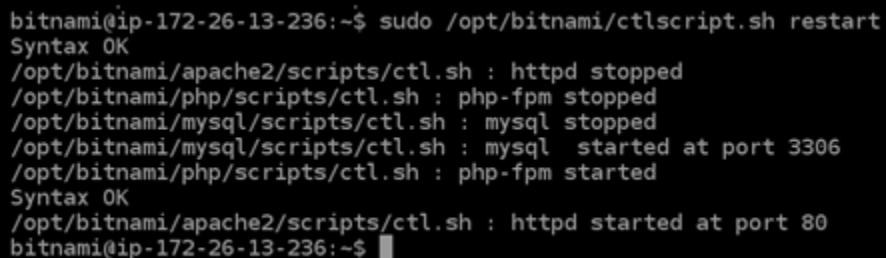
/* That's all, stop editing! Happy blogging. */

define('FS_METHOD', 'direct');
```

6. Press **Ctrl+X** to exit Nano, then press **Y**, and **Enter** to save your edits to the `wp-config.php` file.
7. Enter the following command to restart the services on the instance:

```
sudo /opt/bitnami/ctlscript.sh restart
```

You will see a result similar to the following when the services have restarted:



```
bitnami@ip-172-26-13-236:~$ sudo /opt/bitnami/ctlscript.sh restart
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
/opt/bitnami/php/scripts/ctl.sh : php-fpm stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql started at port 3306
/opt/bitnami/php/scripts/ctl.sh : php-fpm started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80
bitnami@ip-172-26-13-236:~$
```

Close the SSH window and toggle back to the **Offload Media** page that you left open earlier in this tutorial. You are now ready to [create the Amazon S3 bucket using the WP Offload Media plugin](#).

Step 5: Create the Amazon S3 bucket using the WP Offload Media plugin

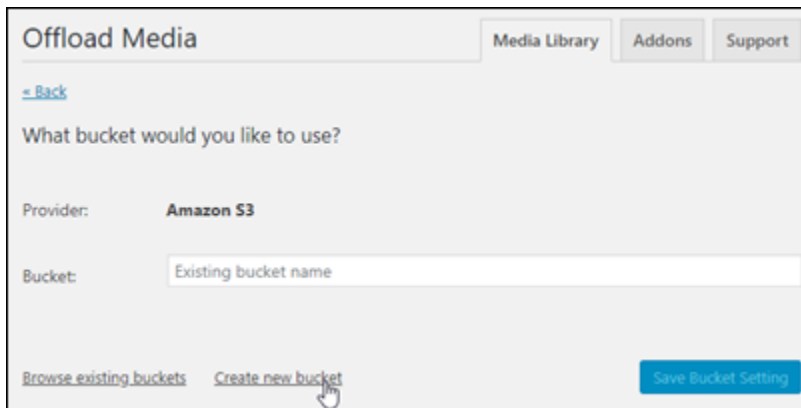
Now that the `wp-config.php` file is configured with the AWS credentials, you can return to the **Offload Media** page to complete the process.

Complete the following steps to create the Amazon S3 bucket using the WP Offload Media plugin.

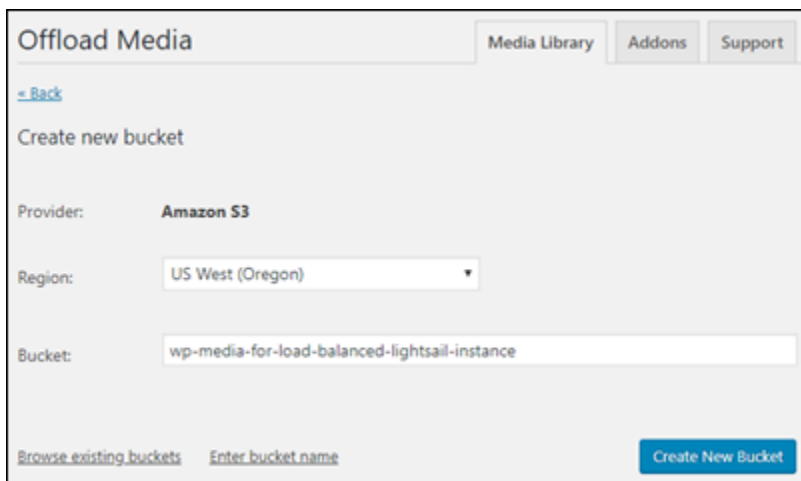
1. Refresh the **Offload Media** page, or choose **Next**.

You should now see that the Amazon S3 provider is configured.

2. Choose **Create new bucket**.

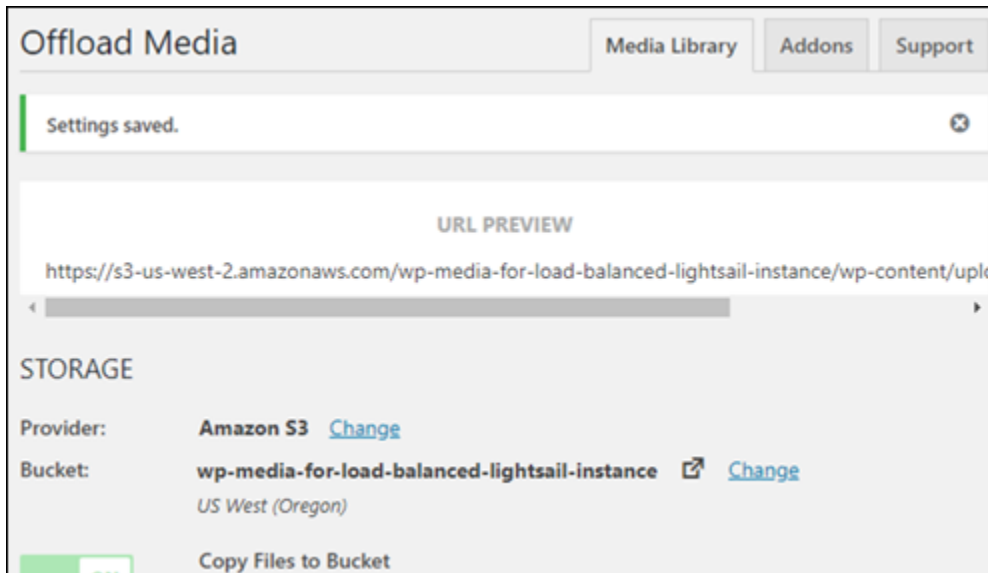


3. In the **Region** drop-down menu, choose the desired AWS Region. We recommend that you choose the same region in which your WordPress instance is located.
4. In the **Bucket** text box, enter a name for the new S3 bucket.



5. Choose **Create New Bucket**.

The page refreshes to confirm that a new bucket was created. Review the settings that appear and adjust them accordingly to how you want your WordPress website to behave.



From now on, images and attachments added to blog posts are automatically uploaded to the Amazon S3 bucket that you created.

Step 6: Next steps

After you're done connecting your WordPress website to an Amazon S3 bucket, you should create a snapshot of your WordPress instance to back up the changes you made. For more information, see [Create a snapshot of your Linux or Unix instance](#).

Tutorial: Connect a WordPress instance in Lightsail to an Amazon Aurora database

Website data for posts, pages, and users is stored on a database that is running on your WordPress instance in Amazon Lightsail. If your instance fails, your data may become unrecoverable. To prevent this scenario, you should transfer your website data to an Amazon Aurora database in the Amazon Relational Database Service (Amazon RDS).

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud. It combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open-source databases. Aurora is offered as part of Amazon RDS. Amazon RDS is a managed database service that makes it easier to set up, operate, and scale a relational database in the cloud. For more information, see the [Amazon Relational Database Service User Guide](#) and the [Amazon Aurora User Guide for Aurora](#).

In this tutorial, we show you how to connect your website database from a WordPress instance in Lightsail to an Aurora managed database in Amazon RDS.

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Configure the security group for your Aurora database](#)
- [Step 3: Connect to your Aurora database from your Lightsail instance](#)
- [Step 4: Transfer the MySQL database from your WordPress instance to your Aurora database](#)
- [Step 5: Configure WordPress to connect to your Aurora managed database](#)

Step 1: Complete the prerequisites

Complete the following prerequisites before you begin:

1. Create a WordPress instance in Lightsail, and configure your application on it. The instance should be in a running state before you continue. For more information, see [Tutorial: Launch and configure a WordPress instance in Amazon Lightsail](#).
2. Turn on VPC peering in your Lightsail account. For more information, see [Set up peering to work with AWS resources outside of Lightsail](#).
3. Create an Aurora managed database in Amazon RDS. The database must be located in the same AWS Region as your WordPress instance. It should also be in a running state before you continue. For more information, see [Getting started with Amazon Aurora](#) in the Amazon Aurora User Guide.

Step 2: Configure the security group for your Aurora database

An AWS security group acts as a virtual firewall for your AWS resources. It controls the incoming and outgoing traffic that can connect to your Aurora database in Amazon RDS. For more information about security groups, see [Control traffic to resources using security groups](#) in the Amazon Virtual Private Cloud User Guide.

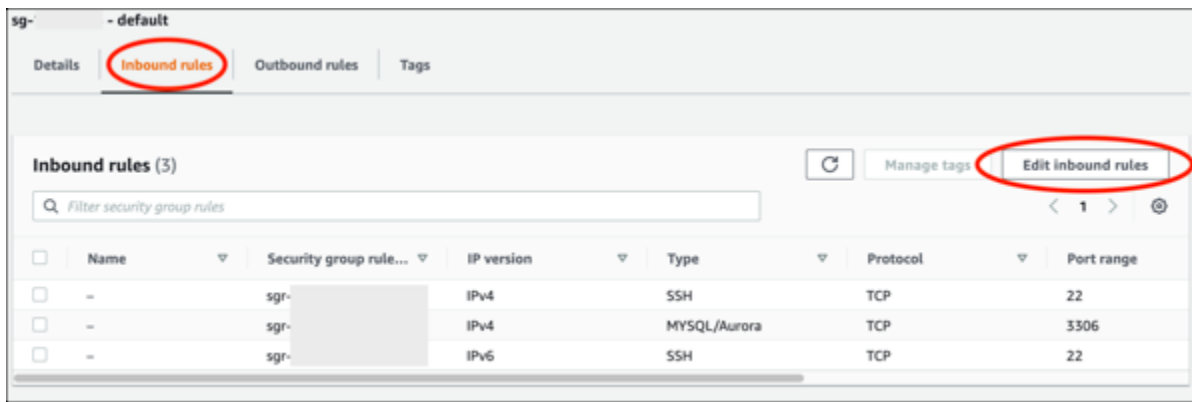
Complete the following procedure to configure the security group so that your WordPress instance can establish a connection to your Aurora database.

1. Sign in to the [Amazon RDS console](#).

2. Choose **Databases** in the navigation pane.
3. Choose the **Writer instance** of the Aurora database that your WordPress instance will connect to.
4. Choose the **Connectivity & security** tab.
5. In the **Endpoint & port** section, make a note of the **Endpoint name** and **Port** of the **Writer instance**. You will need these later when configuring your Lightsail instance to connect to the database.
6. In the **Security** section, choose the active VPC security group link. You will be redirected to your database's security group.

The screenshot displays the AWS Management Console interface for an Aurora database instance. The breadcrumb navigation shows 'RDS > Databases > aurora-database-1 > aurora-database-1-instance-1'. The instance name 'aurora-database-1-instance-1' is at the top, with 'Modify' and 'Actions' buttons. Below is a 'Related' section with a search bar and a table of database instances. The table has columns for DB identifier, Role, Engine, Region & AZ, Size, Status, and CPU. The 'aurora-database-1-instance-1' row is selected, and its 'Writer instance' role is circled in red. Below the table are tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance', and 'Tags'. The 'Connectivity & security' tab is active, showing three sections: 'Endpoint & port', 'Networking', and 'Security'. The 'Endpoint & port' section shows the endpoint 'aurora-database-1-instance-1.us-west-2.rds.amazonaws.com' and the port '3306', both circled in red. The 'Security' section shows 'VPC security groups' with a link to 'default (sg-...)' circled in red, and a status of 'Active'.

7. Make sure that the security group for your Aurora database is selected.
8. Choose the **Inbound rules** tab.
9. Choose **Edit inbound rules**.



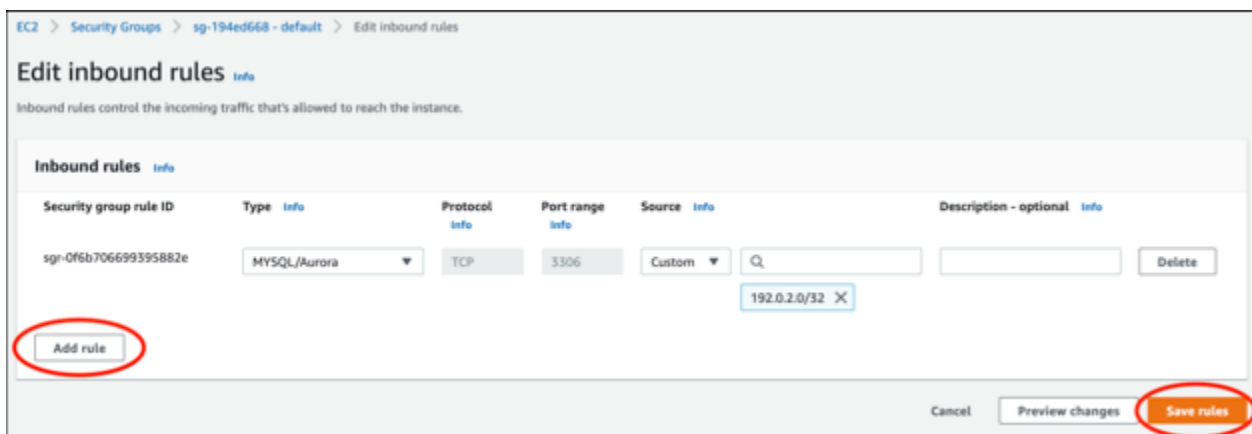
10. In the **Edit inbound rules** page, choose **Add rule**.

11. Complete one of the following steps:

- If you are using the default MySQL port 3306, select **MySQL/Aurora** in the **Type** dropdown menu.
- If you are using a custom port for your database, select **Custom TCP** in the **Type** dropdown menu and enter the port number in the **Port Range** text box.

12. In the **Source** text box, add the private IP address of your WordPress instance. You must enter the IP addresses in CIDR notation, which means that you must append /32. For example, to allow 192.0.2.0, enter 192.0.2.0/32.

13. Choose **Save rules**.

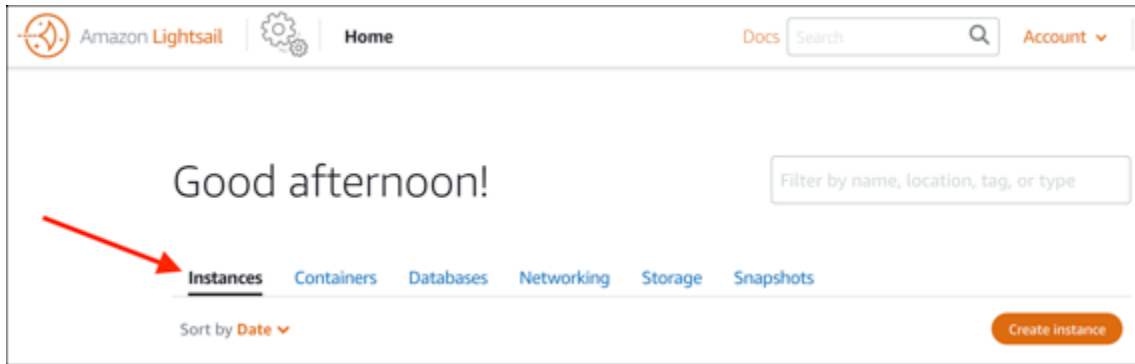


Step 3: Connect to your Aurora database from your Lightsail instance

Complete the following procedure to confirm that you can connect to your Aurora database from your Lightsail instance.

1. Sign in to the [Lightsail console](#).

2. On the Lightsail home page, choose the **Instances** tab.



3. Choose the browser-based SSH client icon for your WordPress instance to connect to it using SSH.



4. After you're connected to your instance, enter the following command to connect to your Aurora database. In the command, replace *DatabaseEndpoint* with the endpoint address of your Aurora database and replace *Port* with the port of your database. Replace *MyUserName* with the name of the user that you entered when creating the database.

```
mysql -h DatabaseEndpoint -P Port -u MyUserName -p
```

You should see a response similar to the following example, which confirms that your instance can access and connect to your Aurora database.

```
bitnami@ip-... $ mysql -h database.cluster-... .us-west-2.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 215
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> █
```

If you don't see this response, or you get an error message, then you might need to configure the security group of your Aurora database to allow the private IP address of your Lightsail instance to connect to it. For more information, see the [Configure the security group for your Aurora database](#) section of this guide.

Step 4: Transfer the database from your WordPress instance to your Aurora database

Now that you've confirmed you can connect to your database from your instance, you should transfer your WordPress website data to your Aurora database.

1. Sign in to the [Lightsail console](#).
2. In the **Instances** tab, choose the browser-based SSH client for your WordPress instance.



3. After the browser-based SSH client is connected to your WordPress instance, enter the following command. The command transfers the data from the `bitnami_wordpress` database that is on your instance and moves it to your Aurora database. In the command, replace *DatabaseUserName* with the name of the primary user that you entered when creating the Aurora database. Replace *DatabaseEndpoint* with the endpoint address of your Aurora database.

```
sudo mysqldump -u root --databases bitnami_wordpress --single-transaction --
compress --order-by-primary -p$(cat /home/bitnami/bitnami_application_password) |
sudo mysql -u DatabaseUserName --host DatabaseEndpoint --password
```

Example

```
sudo mysqldump -u root --databases bitnami_wordpress --single-transaction --
compress --order-by-primary -p$(cat /home/bitnami/bitnami_application_password)
```

```
| sudo mysql -u DBuser --host abc123exampleE67890.czowadgeezqi.us-west-2.rds.amazonaws.com --password
```

4. At the Enter password prompt, enter the password for your Aurora database, and press **Enter**.

You won't be able to see the password while you type it.

```
bitnami@ip-172-26-7-200:~$ mysqldump -u root --databases bitnami_wordpress --single-transaction --compress --order-by-primary -p$(cat /home/bitnami/bitnami_application_password) | mysql -u dbmasteruser --host ls-a3420cc0b7a6b772af722d614e64e5c8298cf01c.czowadgeezqi.us-west-2.rds.amazonaws.com --password
Enter password: mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

If the data transfer succeeds, a response similar to the following example is displayed:

```
Enter password: mysqldump: [Warning] Using a password on the command line interface can be insecure.
bitnami@ip-172-26-7-200:~$
```

If you get an error, confirm that you're using the correct database user name, password, and endpoint, and try again.

Step 5: Configure WordPress to connect to your Aurora database

After you transfer your application data to your Aurora database, you should configure WordPress to connect to it. Complete the following procedure to edit the WordPress configuration file (`wp-config.php`) so that your website connects to your Aurora database.

1. In the browser-based SSH client that is connected to your WordPress instance, enter the following command to create a backup of the `wp-config.php` file:

```
cp /opt/bitnami/wordpress/wp-config.php /opt/bitnami/wordpress/wp-config.php-backup
```

2. Enter the following command to make the `wp-config.php` file writable:

```
sudo chmod 664 /opt/bitnami/wordpress/wp-config.php
```

3. Edit the database user name in the config file to the name of the primary user that you entered when creating the Aurora database.

```
sudo wp config set DB_USER DatabaseUserName
```

- Edit the database host in the config file with the endpoint address and port number of your Aurora database. For example, `abc123exampleE67890.czowadgeezqi.us-west-2.rds.amazonaws.com:3306`.

```
sudo wp config set DB_HOST DatabaseEndpoint:Port
```

- Edit the database password in the config file with the password for your Aurora database.

```
sudo wp config set DB_PASSWORD DatabasePassword
```

- Enter the `wp config list` command to verify that the information you entered in the `wp-config.php` file is correct.

```
sudo wp config list
```

A result similar to the following example appears, displaying your configuration details:

```
bitnami@ip-1]      :~$ sudo wp config list
+-----+-----+-----+
| name   | value                                     | type   |
+-----+-----+-----+
| table_prefix | wp_                                       | variable |
| DB_NAME   | bitnami_wordpress                       | constant |
| DB_USER   | admin                                    | constant |
| DB_PASSWORD | Password1                               | constant |
| DB_HOST   | database.cluster.us-west-2.rds.amazonaws.com:3306 | constant |
+-----+-----+-----+
```

- Enter the following command to restart the web services on your instance:

```
sudo /opt/bitnami/ctlscript.sh restart
```

When the services restart, a result similar to the following example is displayed:

```
bitnami@ip-172-26-13-236:~$ sudo /opt/bitnami/ctlscript.sh restart
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
/opt/bitnami/php/scripts/ctl.sh : php-fpm stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql started at port 3306
/opt/bitnami/php/scripts/ctl.sh : php-fpm started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80
bitnami@ip-172-26-13-236:~$
```

Congratulations! Your WordPress site is now configured to use your Aurora database.

Note

If you need to restore the original `wp-config.php` file, enter the following command to restore it using the backup you created earlier in this tutorial.

```
cp /opt/bitnami/wordpress/wp-config.php-backup /opt/bitnami/wordpress/wp-config.php
```

Tutorial: Connect a WordPress website to a MySQL managed database in Lightsail

Crucial WordPress website data for posts, pages, and users, is stored on the MySQL database that is running on your instance in Amazon Lightsail. If your instance fails, your data may become unrecoverable. To prevent this scenario, you should transfer your website data to a MySQL managed database.

In this tutorial, we show you how to transfer your WordPress website data to a MySQL managed database in Lightsail. We also show you how to edit the WordPress configuration (`wp-config.php`) file on your instance so that your website connects to the managed database, and stops connecting to the database running on the instance.

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Transfer the WordPress database to your MySQL managed database](#)
- [Step 3: Configure WordPress to connect to your MySQL managed database](#)
- [Step 4: Complete the next steps](#)

Step 1: Complete the prerequisites

Complete the following prerequisites before getting started:

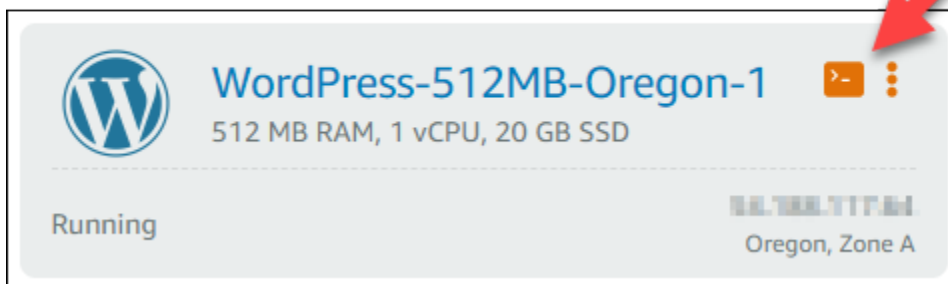
- Create a WordPress instance in Lightsail, and make sure that it's in a running state. For more information, see [Tutorial: Launch and configure a WordPress instance in Amazon Lightsail](#).

- Create a MySQL managed database in Lightsail in the same AWS Region as your WordPress instance, and make sure it's in a running state. WordPress works with all of the MySQL database options available in Lightsail. For more information, see [Creating a database in Amazon Lightsail](#).
- Enable the public mode and data import mode of your MySQL managed database. You can disable these modes after completing the steps in this tutorial. For more information, see [Configure the public mode for your database](#) and [Configure the data import mode for your database](#).

Step 2: Transfer the WordPress database to your MySQL managed database

Complete the following procedure to transfer your WordPress website data to your MySQL managed database in Lightsail.

1. Sign in to the [Lightsail console](#).
2. In the **Instances** tab, choose the browser-based SSH client icon for your WordPress instance.



3. After the browser-based SSH client is connected to your WordPress instance, enter the following command to transfer the data in the `bitnami_wordpress` database that is on your instance to your MySQL managed database. Be sure to replace `DbUserName` with the user name of your managed database, and replace `DbEndpoint` with the endpoint address of your managed database.

```
sudo mysqldump -u root --databases bitnami_wordpress --single-transaction --
compress --order-by-primary -p$(cat /home/bitnami/bitnami_application_password) |
sudo mysql -u DbUserName --host DbEndpoint --password
```

Example

```
sudo mysqldump -u root --databases bitnami_wordpress --single-transaction --
compress --order-by-primary -p$(cat /home/bitnami/bitnami_application_password)
```

```
| sudo mysql -u dbmasteruser --host ls-abc123exampleE67890.czowadgeezi.us-west-2.rds.amazonaws.com --password
```

- At the prompt, enter the password for your MySQL managed database, and press **Enter**.

You will not be able to see the password as it is being typed.

```
bitnami@ip-172-26-7-200:~$ mysqldump -u root --databases bitnami_wordpress --single-transaction --compact --order-by-primary -p$(cat /home/bitnami/bitnami_application_password) | mysql -u dbmasteruser --host ls-a3420cc0b7a6b772af722d614e64e5c8298cf01c.czowadgeezi.us-west-2.rds.amazonaws.com --password
Enter password: mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

- A response similar to the following example is displayed if the data was successfully transferred.

If you get an error, confirm that you're using the correct database user name, password, or endpoint, and try again.

```
Enter password: mysqldump: [Warning] Using a password on the command line interface can be insecure.
bitnami@ip-172-26-7-200:~$
```

Step 3: Configure WordPress to connect to your MySQL managed database

Complete the following procedure to edit the WordPress configuration file (`wp-config.php`) so that your website connects to your MySQL managed database.

- In the browser-based SSH client that is connected to your WordPress instance, enter the following command to create a backup of the `wp-config.php` file in case something goes wrong.

```
cp /opt/bitnami/wordpress/wp-config.php /opt/bitnami/wordpress/wp-config.php-backup
```

- Enter the following command to open the `wp-config.php` file using the Nano text editor.

```
nano /opt/bitnami/wordpress/wp-config.php
```

- Scroll down until you find the values for `DB_USER`, `DB_PASSWORD`, and `DB_HOST` as shown in the following example.

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define('DB_NAME', 'bitnami_wordpress');  
  
/** MySQL database username */  
define('DB_USER', 'bn_wordpress');  
  
/** MySQL database password */  
define('DB_PASSWORD', 'd6ab501583');  
  
/** MySQL hostname */  
define('DB_HOST', 'localhost:3306');
```

4. Modify the following values:

- **DB_USER** — Edit this to match the user name of your MySQL managed database. The default primary user name for Lightsail managed databases is `dbmasteruser`.
- **DB_PASSWORD** — Edit this to match the strong password of your MySQL managed database. For more information, see [Manage your database password](#).
- **DB_HOST** — Edit this to match the endpoint of your MySQL managed database. Be sure to add the `:3306` port number at the end of the host address. For example `ls-abc123exampleE67890.czowadgeezi.us-west-2.rds.amazonaws.com:3306`.

The result should look like the following example.

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define('DB_NAME', 'bitnami_wordpress');  
  
/** MySQL database username */  
define('DB_USER', 'dbmasteruser');  
  
/** MySQL database password */  
define('DB_PASSWORD', 'Q+s) [redacted] ?1|jY');  
  
/** MySQL hostname */  
define('DB_HOST', 'ls-c6d76d20f14d2c [redacted] ca7a695e26.czowadgeezi.us-west-2.rds.amazonaws.com:3306');
```

5. Press **Ctrl+X** to exit Nano, then press **Y** and **Enter** to save your edits.
6. Enter the following command to restart the web services on your instance.

```
sudo /opt/bitnami/ctlscript.sh restart
```

A result similar to the following example is displayed when the services have restarted.


```
bitnami@ip-172-26-13-236:~$ sudo /opt/bitnami/ctlscript.sh restart
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
/opt/bitnami/php/scripts/ctl.sh : php-fpm stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql started at port 3306
/opt/bitnami/php/scripts/ctl.sh : php-fpm started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80
bitnami@ip-172-26-13-236:~$
```

Congratulations! Your WordPress site is now configured to use the MySQL managed database.

Note

If for any reason you need to restore the original `wp-config.php` file, enter the following command to restore it using the backup you created earlier in this tutorial.

```
cp /opt/bitnami/wordpress/wp-config.php-backup /opt/bitnami/wordpress/wp-config.php
```

Step 4: Complete the next steps

You should complete these additional steps after you're done connecting your WordPress website to a MySQL managed database:

- Create a snapshot of your WordPress instance. For more information, see [Create a snapshot of your Linux or Unix instance](#).
- Create a snapshot of the MySQL managed database. For more information, see [Create a snapshot of your database](#).
- Disable the public mode and data import mode of your MySQL managed database. For more information, see [Configure the public mode for your database](#) and [Configure the data import mode for your database](#).

Tutorial: Connect a WordPress instance to a Lightsail bucket

This tutorial describes the steps required to connect your WordPress website running on an Amazon Lightsail instance to a Lightsail bucket. You can use the bucket to host static content

such as images and attachments. To do this, you must install the WP Offload Media Lite plugin on your WordPress website and configure it to connect to your Lightsail bucket. After the plugin is configured, all media that you upload to your WordPress website is automatically added to your bucket instead of the instance's disk.

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Modify your bucket permissions](#)
- [Step 3: Install the WP Offload Media Lite plugin on your WordPress website](#)
- [Step 4: Test the connection between your WordPress website and your Lightsail bucket](#)

Step 1: Complete the prerequisites

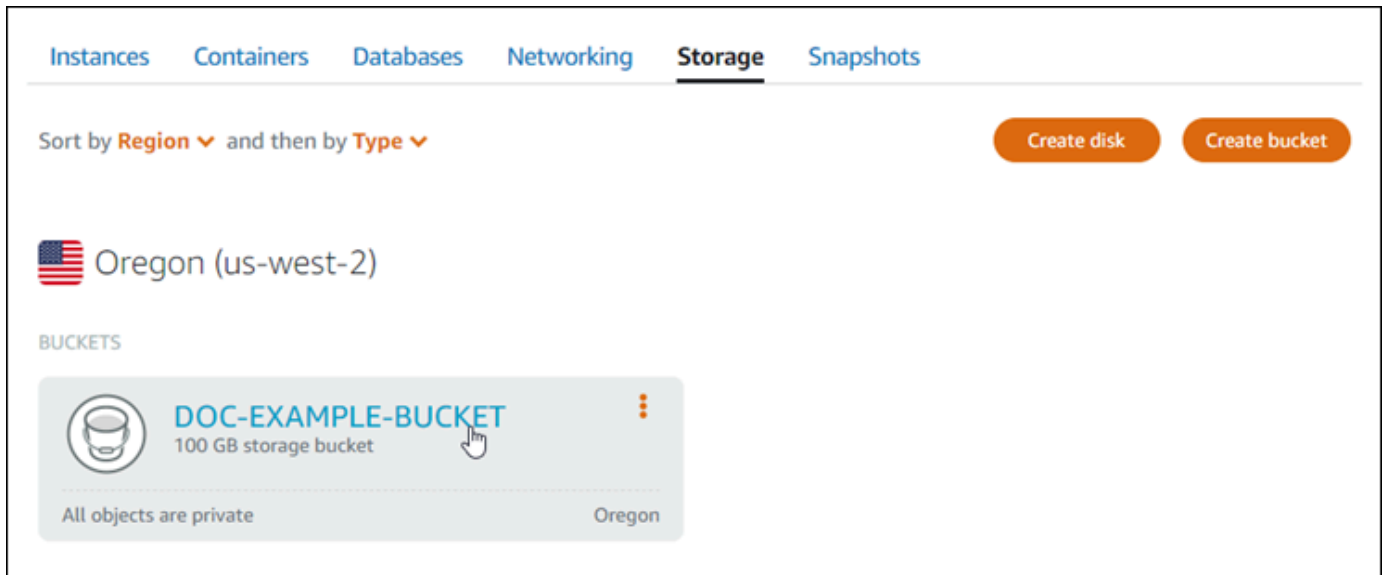
Complete the following prerequisites if you haven't already:

- Create a WordPress instance in Lightsail. For more information, see [Tutorial: Launch and configure a WordPress instance in Amazon Lightsail](#).
- Create a bucket in the Lightsail object storage service. For more information, see [Create a bucket](#).

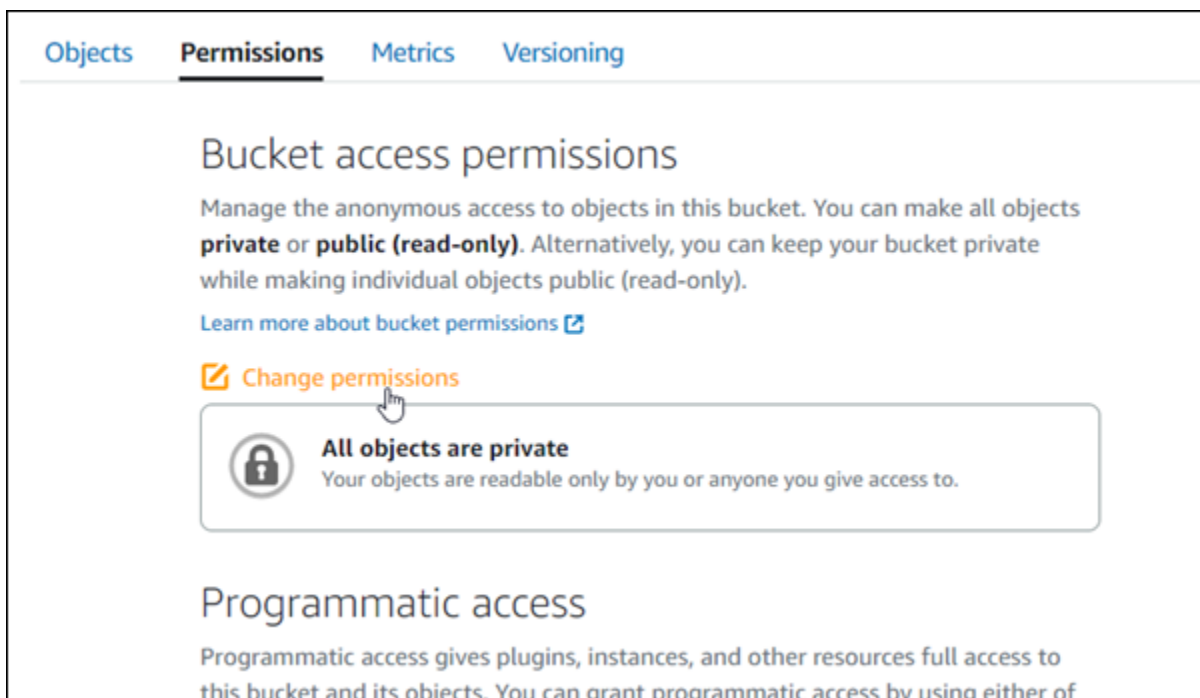
Step 2: Modify your bucket permissions

Complete the following procedure to change the permissions of your bucket to give access to your WordPress instance and the Offload Media Lite plugin. The access permissions of your bucket must be set to **Individual objects can be made public (read-only)**. You must also attach the WordPress instance to the access role of your bucket. For more information about bucket permissions, see [Bucket permissions](#).

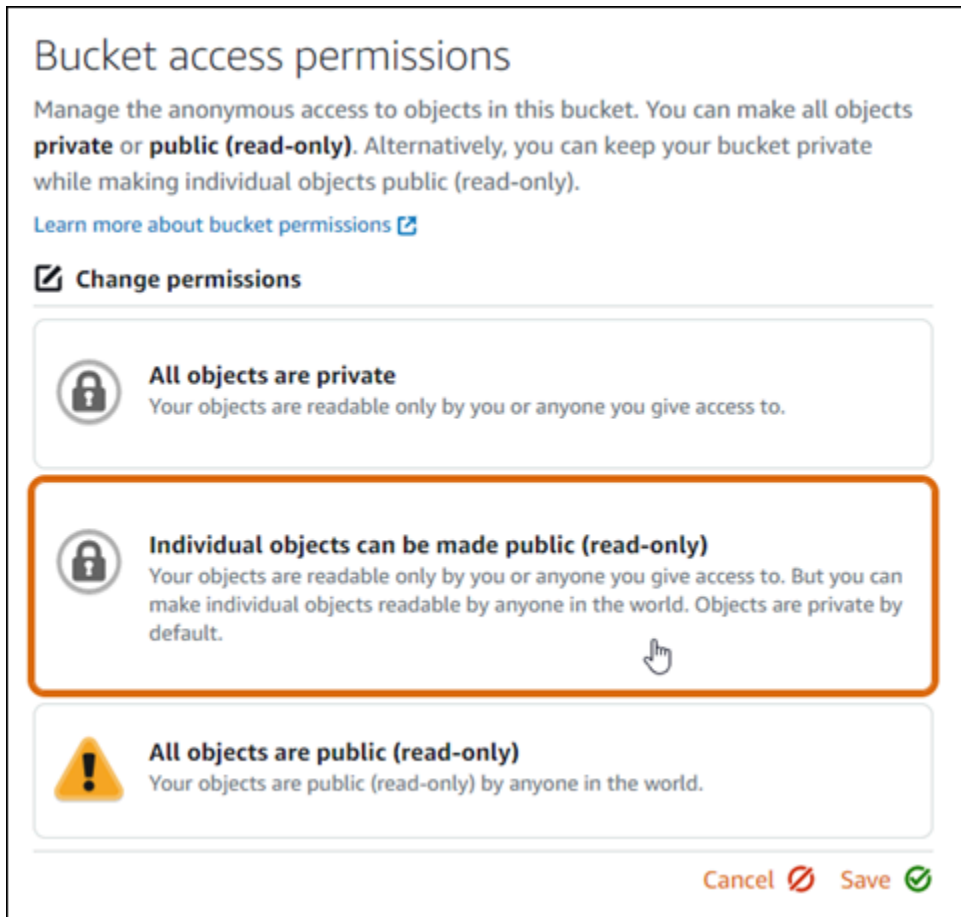
1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket that you want to use with your WordPress website.



4. Choose the **Permissions** tab on the **Bucket management** page.
5. Choose **Change permissions** under the **Bucket access permissions** section of the page.



6. Choose **Individual objects can be made public and read only**.



Bucket access permissions

Manage the anonymous access to objects in this bucket. You can make all objects **private** or **public (read-only)**. Alternatively, you can keep your bucket private while making individual objects public (read-only).

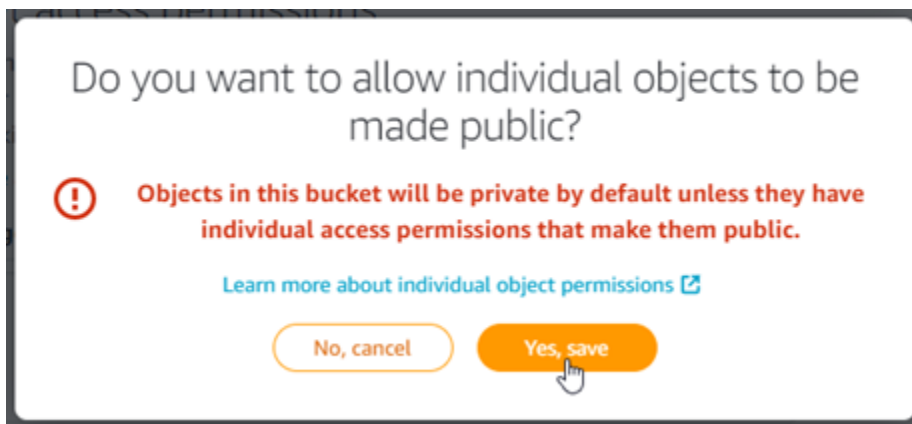
[Learn more about bucket permissions](#)

Change permissions

- All objects are private**
Your objects are readable only by you or anyone you give access to.
- Individual objects can be made public (read-only)**
Your objects are readable only by you or anyone you give access to. But you can make individual objects readable by anyone in the world. Objects are private by default.
- All objects are public (read-only)**
Your objects are public (read-only) by anyone in the world.

Cancel Save

7. Choose **Save**.
8. Choose **Yes, save** in the confirmation prompt that appears.



Do you want to allow individual objects to be made public?

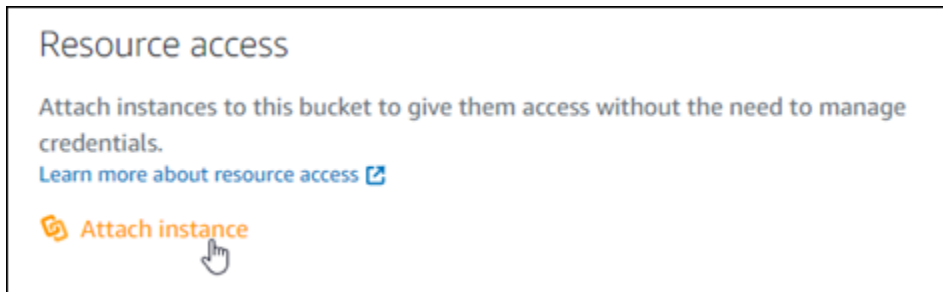
! Objects in this bucket will be private by default unless they have individual access permissions that make them public.

[Learn more about individual object permissions](#)

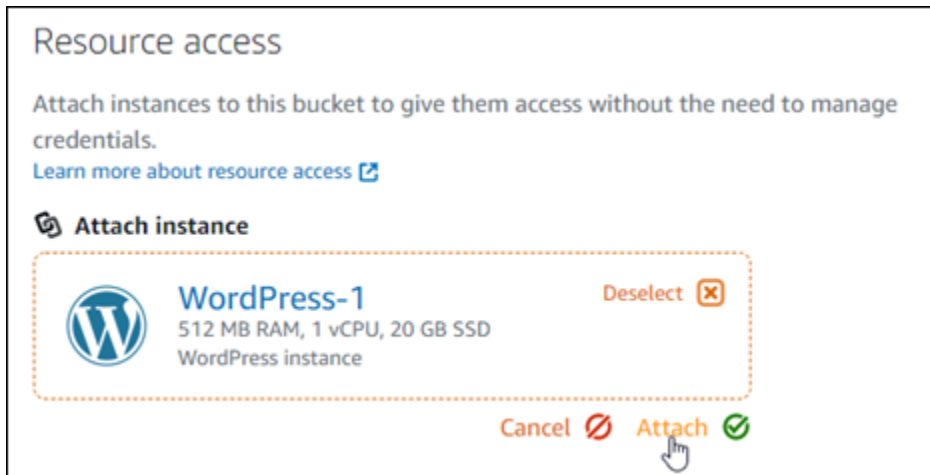
No, cancel Yes, save

After a few moments, your bucket is configured to allow for individual object access. This ensures that objects uploaded to your bucket from your WordPress website using the Offload Media Lite plugin are readable to your customers.

9. Scroll to the **Resource access** section of the page, and choose **Attach instance**.



- Choose the name of your WordPress instance in the drop-down list that appears, and then choose **Attach**.



After a few moments, your WordPress instance is attached to your bucket. This gives your WordPress instance access to manage your bucket and its objects.

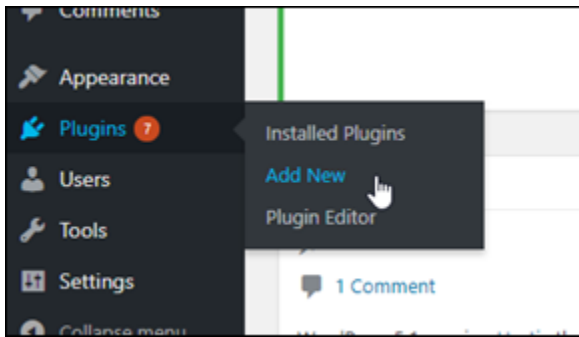
Step 3: Install the WP Offload Media Lite plugin on your WordPress website

Complete the following procedure to install the WP Offload Media Lite plugin on your WordPress website. This plugin automatically copies images, videos, documents, and any other media added through the WordPress media uploader to your Lightsail bucket. For more information, see [WP Offload Media Lite](#) in the *WordPress website*.

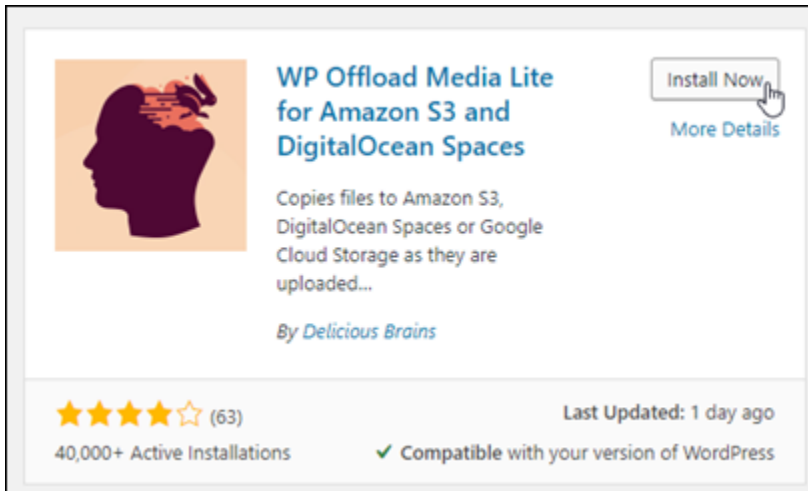
- Sign in to the dashboard of your WordPress website as an administrator.

For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

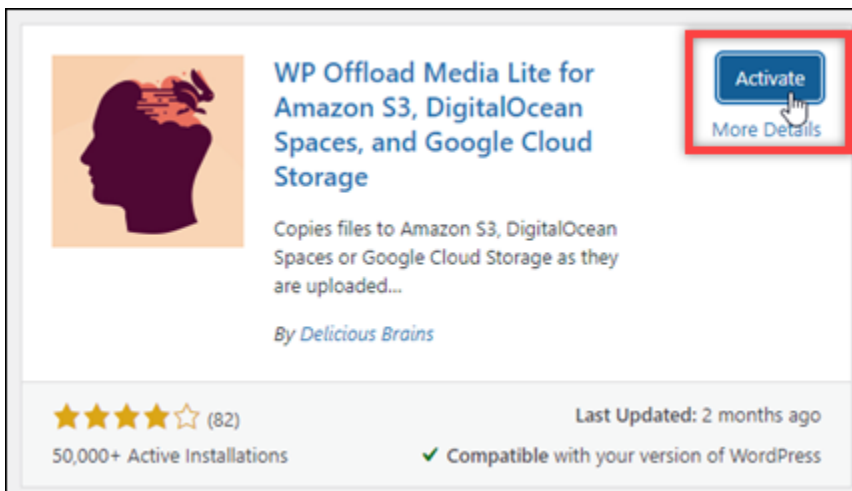
- Pause on **Plugins** in the left navigation menu, and choose **Add New**.



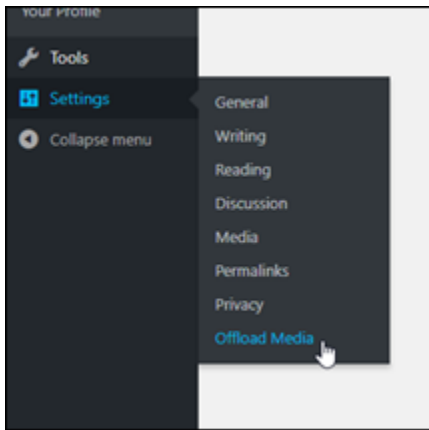
3. Search for **WP Offload Media Lite**.
4. In the search results, choose **Install Now** next to the **WP Offload Media** plugin.



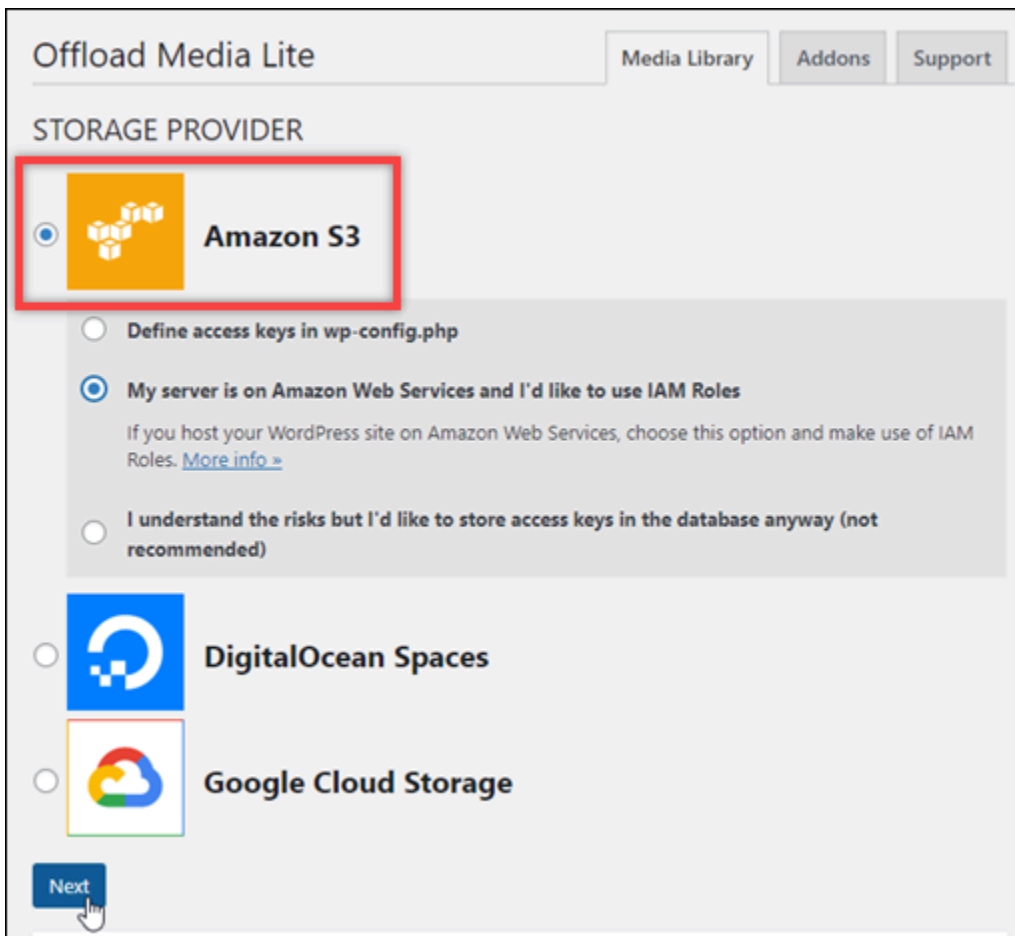
5. Choose **Activate** after the plugin is done installing.



6. In the left navigation menu, choose **Settings**, and then choose **Offload Media**.




7. In the **Offload Media** page, choose **Amazon S3** as the storage provider.



8. Choose **My server is on Amazon Web Services and I'd like to use IAM Roles**.

Offload Media Lite Media Library Addons Support


STORAGE PROVIDER


 **Amazon S3**

Define access keys in wp-config.php

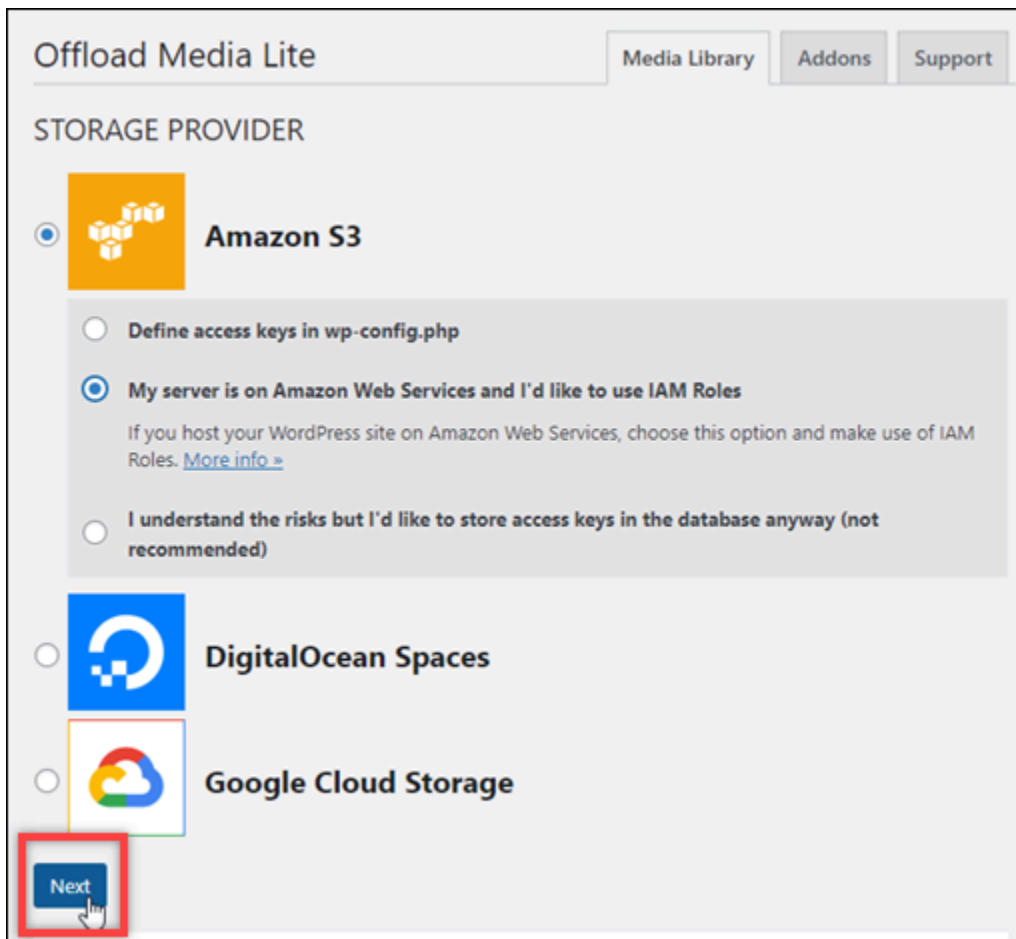
My server is on Amazon Web Services and I'd like to use IAM Roles
If you host your WordPress site on Amazon Web Services, choose this option and make use of IAM Roles. [More info >](#)

I understand the risks but I'd like to store access keys in the database anyway (not recommended)

 **DigitalOcean Spaces**

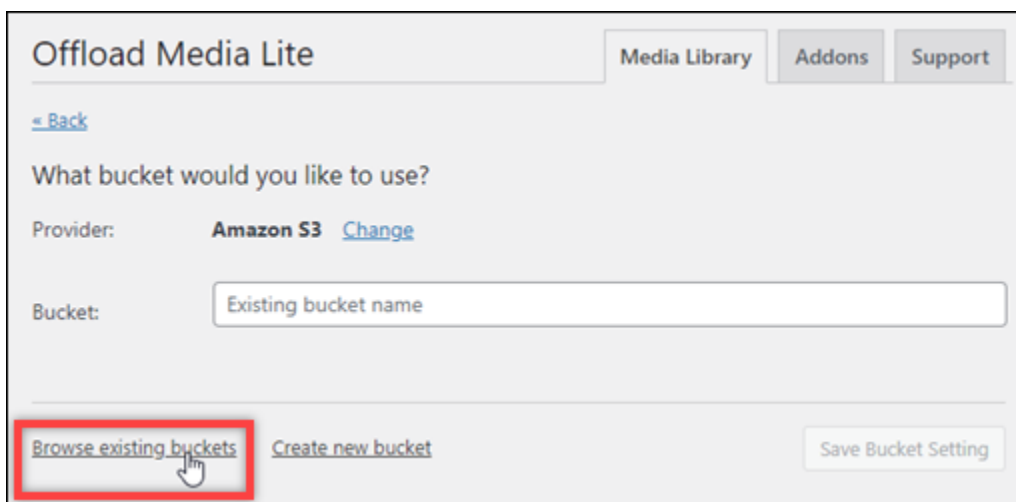
 **Google Cloud Storage**

9. Choose **Next**.



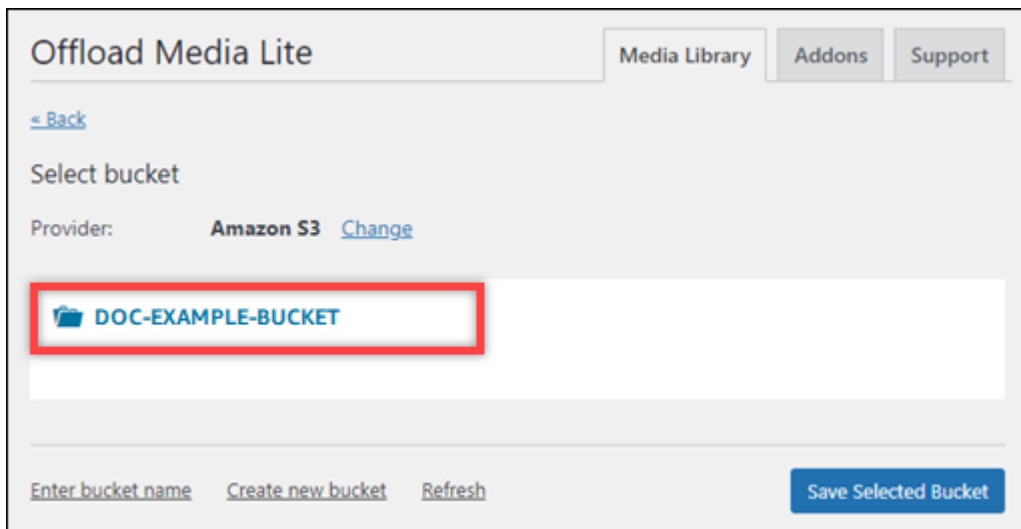
The screenshot shows the 'Offload Media Lite' configuration interface. At the top, there are navigation tabs for 'Media Library', 'Addons', and 'Support'. Below the title, the section is titled 'STORAGE PROVIDER'. Three options are listed: 'Amazon S3' (selected with a radio button), 'DigitalOcean Spaces', and 'Google Cloud Storage'. Under 'Amazon S3', there are three radio button options: 'Define access keys in wp-config.php', 'My server is on Amazon Web Services and I'd like to use IAM Roles' (selected), and 'I understand the risks but I'd like to store access keys in the database anyway (not recommended)'. A 'Next' button is highlighted with a red box at the bottom left.

10. Choose **Browse existing buckets** in the **What bucket would you like to use?** page that appears.



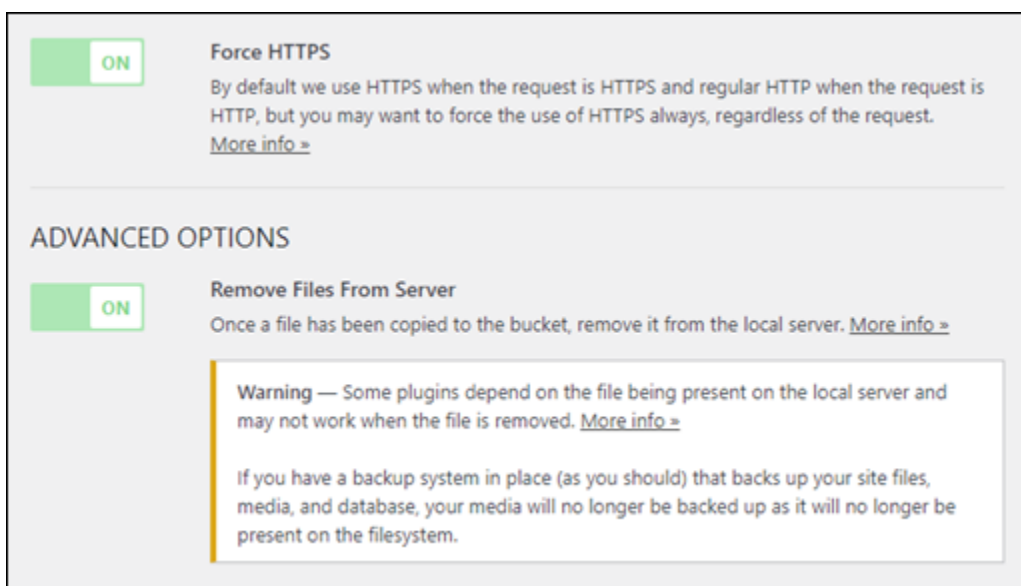
The screenshot shows the 'What bucket would you like to use?' configuration page. At the top, there are navigation tabs for 'Media Library', 'Addons', and 'Support'. Below the title, there is a '< Back' link. The main heading is 'What bucket would you like to use?'. Below this, the 'Provider' is set to 'Amazon S3' with a 'Change' link. The 'Bucket' field contains the text 'Existing bucket name'. At the bottom, there are three buttons: 'Browse existing buckets' (highlighted with a red box), 'Create new bucket', and 'Save Bucket Setting'.

11. Choose the name of the bucket that you want to use with your WordPress instance.



12. In the **Offload Media Lite Settings** page that appears, make sure to turn on **Force HTTPS** and **Remove Files From Server**.

- The **Force HTTPS** setting must be turned on because Lightsail buckets use HTTPS by default to serve media files. If you don't turn this feature on, media files that are uploaded to your Lightsail bucket from your WordPress website won't be served correctly to your website visitors.
- The **Remove Files From Server** setting ensures that media that is uploaded to your Lightsail bucket isn't also stored on your instance's disk. If you don't turn this feature on, media files that are uploaded to your Lightsail bucket are also stored on the local storage of your WordPress instance.



13. Choose **Save Changes**.

Note

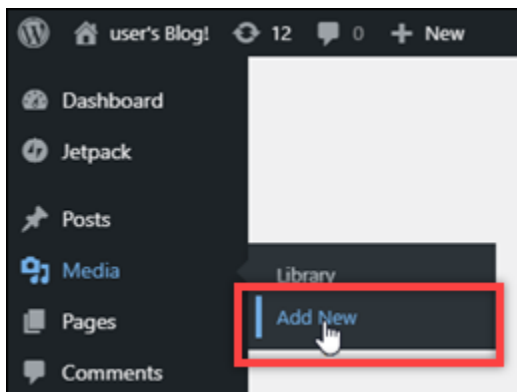
To return to the **Offload Media Lite Settings** page later, pause on **Settings** in the left navigation menu, and choose **Offload Media Lite**.

Your WordPress website is now configured to use the Media Lite Plugin. The next time you upload a media file through WordPress, that file is automatically uploaded to your Lightsail bucket, and is served by the bucket. To test the configuration, continue to the next section of this tutorial.

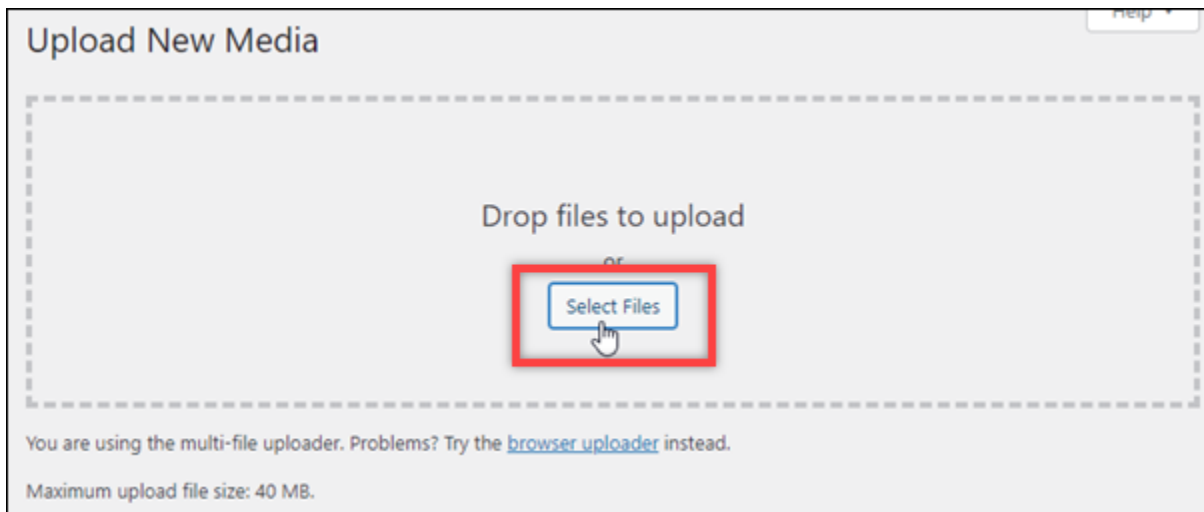
Step 4: Test the connection between your WordPress website and your Lightsail bucket

Complete the following procedure to upload a media file to your WordPress instance and confirm that it is uploaded to, and is served from your Lightsail bucket.

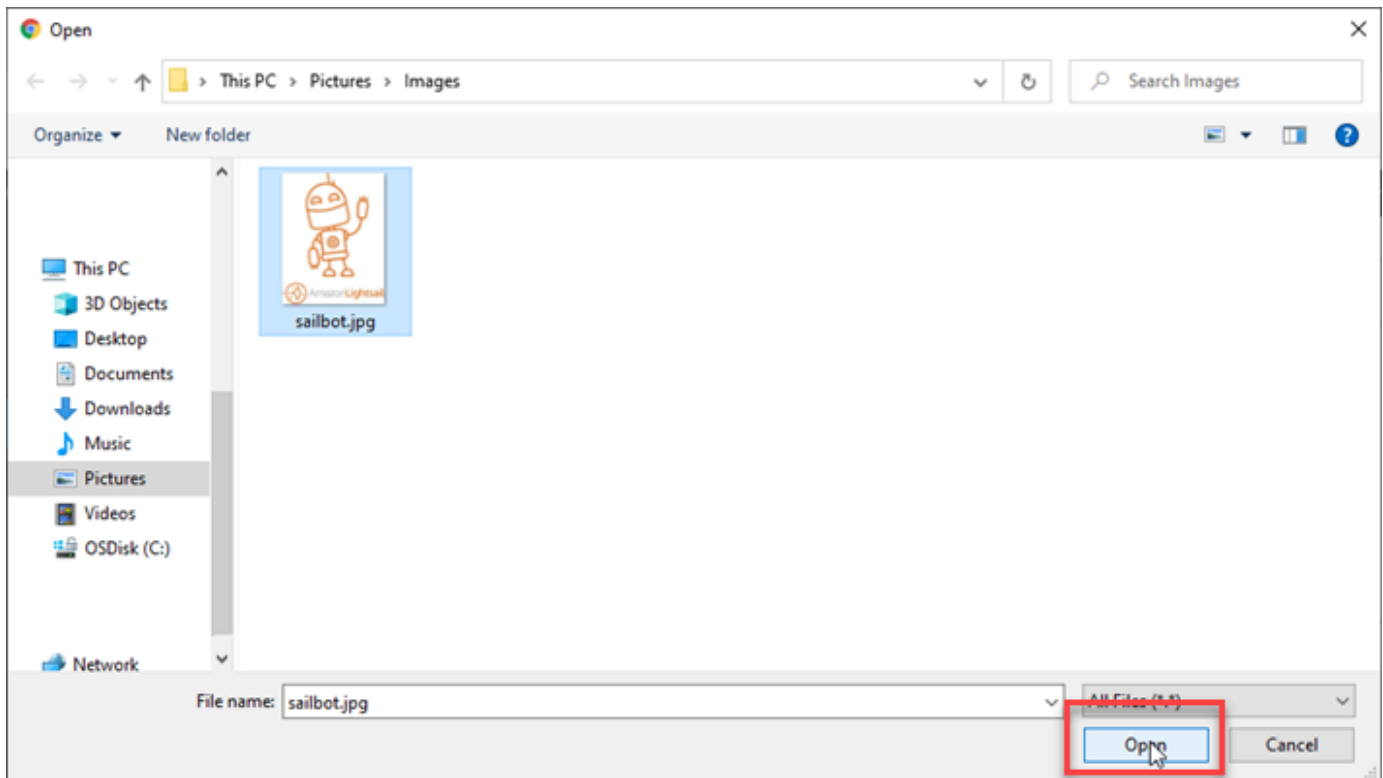
1. Pause on **Media** in the left navigation menu of the WordPress dashboard, and choose **Add New**.



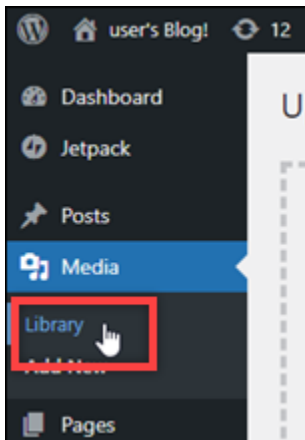
2. Choose **Select Files** on the Upload New Media page that appears.



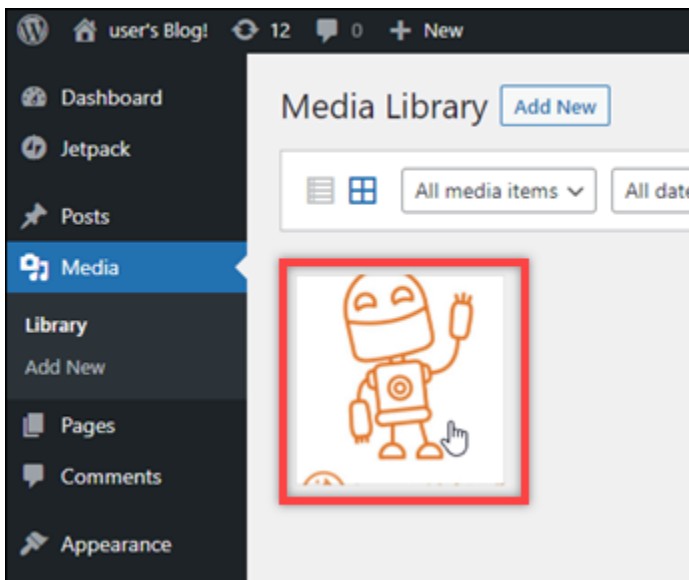
3. Choose a media file to upload from your local computer, and choose **Open**.



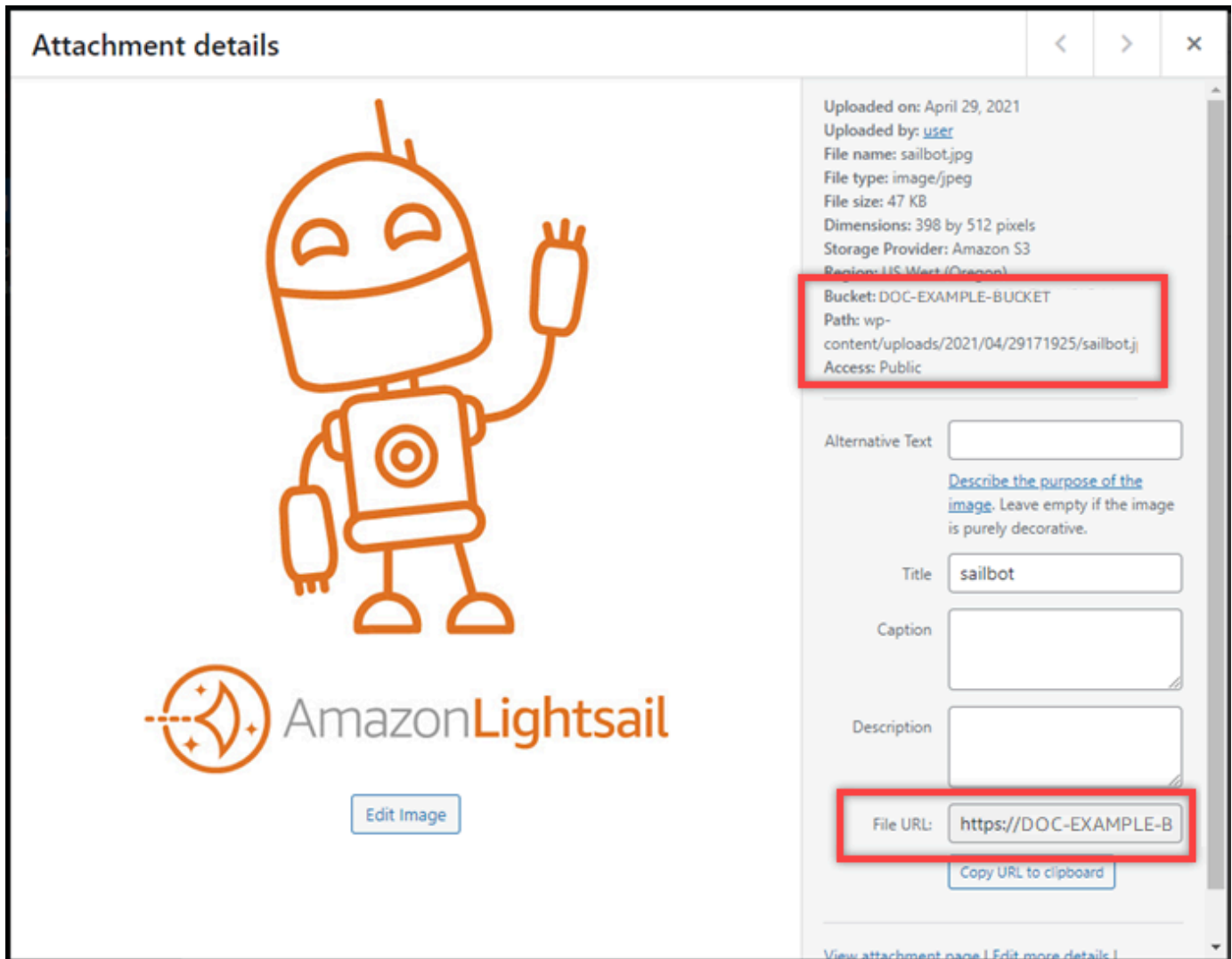
4. When the file is done uploading, choose **Library** under **Media** in the left navigation menu.



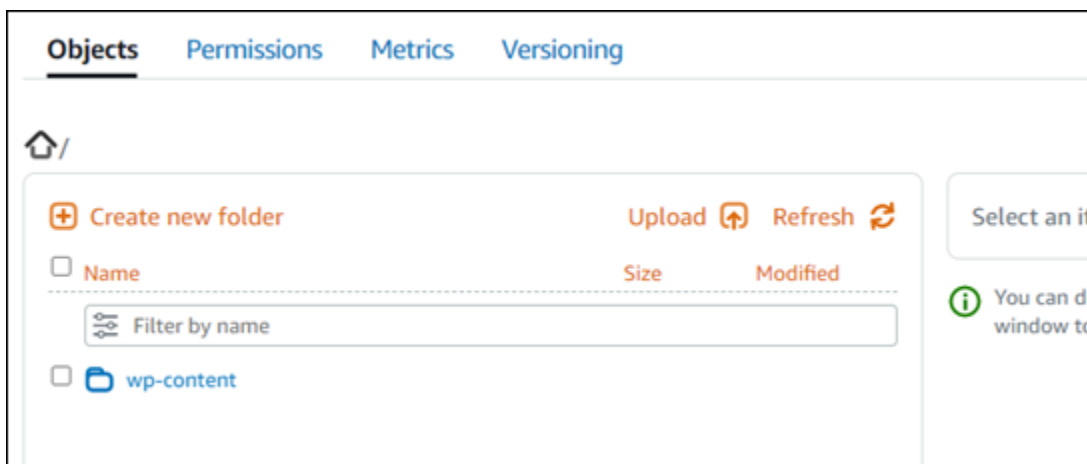
5. Choose the file that you recently uploaded.



6. In the details panel of the file, you should see the name of your bucket in the **Bucket** and **File URL** fields.



- When you go to the **Objects** tab of the Lightsail bucket management page, you should see a **wp-content** folder. This folder is created by the Offload Media Lite plugin and is used to store your uploaded media files.



Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
 6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).

7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
14. Learn how to connect your bucket to other resources. For more information, see the following tutorials.
 - [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
15. Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Configure your WordPress instance to work with a content delivery network distribution in Lightsail

In this guide, we show you how to configure your WordPress instance to work with a Amazon Lightsail distribution.

All Lightsail distributions have HTTPS enabled by default for their default domain (for example, 123456abcdef.cloudfront.net). The configuration of your distribution determines whether the connection between your distribution and your instance is encrypted.

- **Your WordPress website uses HTTP only** – If your website uses HTTP only as the origin of your distribution, and it is not configured to use HTTPS, you can configure your distribution to terminate SSL/TLS and forward all content requests to your instance using an unencrypted connection.
- **Your WordPress website uses HTTPS** – If your website uses HTTPS as the origin of your distribution, you can configure your distribution to forward all content requests to your instance using an encrypted connection. This configuration is known as end-to-end encryption.

Create the distribution

Complete the following steps to configure a Lightsail distribution for your WordPress instance. For more information, see [the section called “Create a distribution”](#).

Prerequisite

Create and configure a WordPress instance as described in [the section called “WordPress”](#).

To create a distribution for your WordPress instance

1. On the Lightsail home page, choose **Networking**.
2. Choose **Create distribution**.
3. For **Choose your origin**, choose the Region where you're running your WordPress instance and then choose your WordPress instance. We automatically use the static IP address that you attached to the instance.
4. For **Caching behavior**, choose **Best for WordPress**.
5. (Optional) To configure end-to-end encryption, change the origin protocol policy to **HTTPS only**. For more information, see [the section called “Origin protocol policy”](#).

6. Configure the remaining options and then choose **Create distribution**.
7. On the **Custom domains** tab, choose **Create certificate**. Enter a unique name for the certificate, enter the names of your domain and subdomains, and then choose **Create certificate**.
8. Choose **Attach certificate**.
9. For **Update DNS records**, choose **I understand**.

Update DNS records

Complete the following steps to update the DNS records for your Lightsail DNS zone.

To update the DNS records for your distribution

1. On the Lightsail home page, choose **Domains & DNS**.
2. Choose your DNS zone and then choose the **DNS records** tab.
3. Delete the A and AAAA records for the domain that you specified in your certificate.
4. Choose **Add record** and create a CNAME record that resolves your domain to the domain for your distribution (for example, d2vbec9EXAMPLE.cloudfront.net).
5. Choose **Save**.

Allow static content to be cached by the distribution

Complete the following procedure to edit the `wp-config.php` file in your WordPress instance so that it works with your distribution.

Note

We recommend that you create a snapshot of your WordPress instance before getting started with this procedure. The snapshot can be used as a backup from which you can create another instance in case something goes wrong. For more information, see [Create a snapshot of your Linux or Unix instance](#).

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the browser-based SSH client icon that is displayed next to your WordPress instance.

3. After you're connected to your instance, enter the following command to create a backup of the `wp-config.php` file. If something goes wrong, you can restore the file using the backup.

```
sudo cp /opt/bitnami/wordpress/wp-config.php /opt/bitnami/wordpress/wp-config.php.backup
```

4. Enter the following command to open the `wp-config.php` file using Vim.

```
sudo vim /opt/bitnami/wordpress/wp-config.php
```

5. Press **I** to enter insert mode in Vim.
6. Delete the following lines of code in the file.

```
define('WP_SITEURL', 'http://' . $_SERVER['HTTP_HOST'] . '/');  
define('WP_HOME', 'http://' . $_SERVER['HTTP_HOST'] . '/');
```

7. Add one of the following lines of code to the file depending on the version of WordPress that you're using:

- If you're using version 3.3 or lower, add the following lines of code where you previously deleted the code.

```
define('WP_SITEURL', 'https://' . $_SERVER['HTTP_HOST'] . '/');  
define('WP_HOME', 'https://' . $_SERVER['HTTP_HOST'] . '/');  
if (isset($_SERVER['HTTP_CLOUDFRONT_FORWARDED_PROTO'])  
&& $_SERVER['HTTP_CLOUDFRONT_FORWARDED_PROTO'] === 'https') {  
    $_SERVER['HTTPS'] = 'on';  
}
```

- If you're using version 3.3.1-5 or higher, add the following lines of code where you previously deleted the code.

```
define('WP_SITEURL', 'http://DOMAIN/');  
define('WP_HOME', 'http://DOMAIN/');  
if (isset($_SERVER['HTTP_CLOUDFRONT_FORWARDED_PROTO'])  
&& $_SERVER['HTTP_CLOUDFRONT_FORWARDED_PROTO'] === 'https') {  
    $_SERVER['HTTPS'] = 'on';  
}
```

8. Press the **Esc** key to exit insert mode in Vim, then type `:wq!` and press **Enter** to save your edits (write) and quit Vim.

9. Enter the following command to restart the Apache service on your instance.

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

10. Wait a few moments for your the Apache service to restart, then test that your distribution is caching your content. For more information, see [Test your Amazon Lightsail distribution](#).

11. If something went wrong, re-connect to your instance using the browser-based SSH client. Run the following command to restore the `wp-config.php` file using the backup you created earlier in this guide.

```
sudo cp /opt/bitnami/wordpress/wp-config.php.backup /opt/bitnami/wordpress/wp-config.php
```

After you restore the file, enter the following command to restart the Apache service:

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

Additional information about distributions

Here are some articles to help you manage distributions in Lightsail:

- [Content delivery network distributions](#)
- [Creating distributions](#)
- [Understand request and response behaviors of a distribution](#)
- [Test your distribution](#)
- [Change the origin of your distribution](#)
- [Change the caching behavior of your distribution](#)
- [Reset the cache of your distribution](#)
- [Change the plan of your distribution](#)
- [Enable custom domains for your distribution](#)
- [Point your domains to your distribution](#)
- [Change custom domains for your distribution](#)
- [Disable custom domains for your distributions](#)
- [View distribution metrics](#)

- [Delete your distribution](#)

Enable email on your WordPress instance in Lightsail

You can enable email on your WordPress instance in Amazon Lightsail. Configure the SMTP service in the Amazon Simple Email Service (Amazon SES). Then activate and configure the WP Mail SMTP plugin on your instance. After email is enabled, your WordPress administrators can request password resets for their user profiles, and will be sent email notifications for blog posts, website updates, and other plugin messages. This guide shows you how to enable email on your WordPress instance in Amazon Lightsail using Amazon SES.





Contents

- [Step 1: Review the restrictions](#)
- [Step 2: Complete the prerequisites](#)
- [Step 3: Create SMTP credentials in Amazon SES](#)
- [Step 4: Verify your domain in Amazon SES](#)
- [Step 5: Verify email addresses in Amazon SES](#)
- [Step 6: Configure the WP Mail SMTP plugin on your WordPress instance](#)

For more information, see [Using the Amazon SES SMTP Interface to Send Email](#) in the Amazon SES documentation.

Step 1: Review the restrictions

New Amazon Web Services (AWS) accounts that are in the Amazon SES sandbox can send email only to verified addresses and domains. If this is the case for your account, then we recommend that you verify your website's domain, and verify the email addresses of your WordPress administrators. To get their email addresses, sign in to your WordPress website's dashboard, and choose **Users** in the left-navigation menu. You'll see the administrator email addresses listed in the **Email** column as shown in the following example:

<input type="checkbox"/> Username	Name	Email	Role
<input type="checkbox"/>  Carlos	Carlos Salazar	user1@lightsail-demo.com	Administrator
<input type="checkbox"/>  Jane	Jane Doe	user2@lightsail-demo.com	Administrator
<input type="checkbox"/>  John	John Doe	user3@lightsail-demo.com	Administrator
<input type="checkbox"/>  user	—	user@example.com	Administrator

Note

The default user profile is configured with the `user@example.com` email address. You should change this to a working email address. For more information, see [Users Profile Screen](#) in the WordPress documentation.

To send email to any address and domain, you must request to have your account taken out of the Amazon SES sandbox. For more information, see [Moving Out of the Amazon SES Sandbox](#) in the Amazon SES documentation.

Step 2: Complete the prerequisites

You must complete the following tasks before you can enable email on your WordPress instance:

- Create a WordPress instance in Lightsail. For more information, see [Tutorial: Launch and configure a WordPress instance in Amazon Lightsail](#).
- Point your registered domain to your WordPress instance using a Lightsail DNS zone. For more information, see [Create a DNS zone to manage your domain's DNS records](#).
- Sign up for Amazon SES and learn more about the service. For more information about signing up for Amazon SES, see [Amazon SES Quick Start](#) in the Amazon SES documentation. For more information about Amazon SES, see the following guides in the Amazon SES documentation:
 - [Amazon SES Developer Guide](#)
 - [Amazon SES FAQs](#)
 - [Amazon SES Pricing](#)
 - [Amazon SES Service Quotas](#)

Step 3: Create SMTP credentials in Amazon SES

Creating SMTP credentials in your Amazon SES account is required to configure the WP Mail SMTP plugin that you configure later in this guide. For more information, see [Obtaining Your Amazon SES SMTP Credentials](#) in the Amazon SES documentation.

To create SMTP credentials in Amazon SES

1. Sign in to the [Amazon SES console](#).
2. From the left-navigation menu, choose **SMTP settings**.

The **SMTP settings** page displays your SMTP server name, ports, and TLS setting. Note these values because you need them later in this guide when configuring the WP Mail SMTP plugin on your WordPress instance.

Server Name:	email-smtp.us-west-2.amazonaws.com
Port:	25, 465 or 587
Use Transport Layer Security (TLS):	Yes
Authentication:	Your SMTP credentials. See below for more information.

3. Choose **Create SMTP credentials**.
4. In the **IAM User Name** text box, leave the default user name, then choose **Create**.

This form lets you create an IAM user for SMTP authentication with Amazon SES. The default user name is `ses-smtp-user-XXXXXXXXXX`. Click **Create** to set up your SMTP credentials.

IAM User Name: (Maximum 64 characters)

[▶ Show More Information](#)

5. Choose **Show User SMTP Security Credentials** to view the SMTP username and password, or choose **Download Credentials** to download a CSV file containing the same information. You need these credentials later when configuring the WP Mail SMTP plugin on your WordPress instance.

▼ Hide User SMTP Security Credentials

ses-smtp-user-XXXXXXXXXX

SMTP Username: AKIA-XXXXXXXXXX-E6QVP

SMTP Password: BLIPyr-XXXXXXXXXX-jSYstFEPtnPp

Note

The credentials created in the Amazon SES console are automatically added to AWS Identity and Access Management (IAM) for your account.

Step 4: Verify your domain in Amazon SES

Amazon SES requires that you verify your domain to confirm that you own it and to prevent others from using it. When you verify a domain, you are verifying all email addresses from that domain, so you don't need to verify email addresses from that domain individually. For example, if you verify the domain `example.com`, you can send email from `user1@example.com`, `user2@example.com`, or any other user at `example.com`. For more information, see [Verifying Domains in Amazon SES](#) in the Amazon SES documentation.

To verify your domain in Amazon SES

1. In the [Amazon SES console](#), from the left-navigation menu, choose **Verified identities**.
2. Choose **Create identity**.
3. Enter the domain that you want to verify, and choose **Create identity**.

The domain that you verify should be the same domain that you're using with your WordPress instance in Lightsail.

Important

Legacy TXT records

Domain verification in Amazon SES is now based on DomainKeys Identified Mail (DKIM), an email authentication standard that receiving mail servers use to validate an email's authenticity. Configuring DKIM in your domain's DNS settings confirms to SES that you're the identity owner, eliminating the need for TXT records. Domain identities that were verified using TXT records do not need to be reverified; however, we still recommend enabling DKIM signatures to enhance the deliverability of your mail with DKIM-compliant email providers.

Create identity

A *verified identity* is a domain, subdomain, or email address you use to send email through Amazon SES. Identity verification at the domain level extends to all email addresses under one verified domain identity.

Identity details [Info](#)

Identity type

Domain

To verify ownership of a domain, you must have access to its DNS settings to add the necessary records.

Email address

To verify ownership of an email address, you must have access to its inbox to open the verification email.

Domain

Domain name can contain up to 253 alphanumeric characters.

Assign a default configuration set

Enabling this option ensures that the assigned configuration set is applied to messages sent from this identity by default whenever a configuration set isn't specified at the time of sending.

Use a custom MAIL FROM domain

Configuring a custom MAIL FROM domain for messages sent from this identity enables the MAIL FROM address to align with the From address. Domain alignment must be achieved in order to be DMARC compliant.

Verifying your domain

DKIM-based domain verification

DomainKeys Identified Mail (DKIM) is an email authentication method that Amazon SES uses to verify domain ownership and that receiving mail servers use to validate email authenticity. You must configure DKIM as part of the domain verification process.

Configuring DKIM

Following identity creation, Amazon SES will provide a set of DNS records. These records must be published to your domain's DNS server in order to successfully configure DKIM and verify ownership of your domain. For more information, see [Verifying a domain with Amazon SES](#).

i If your domain is registered with **Amazon Route 53**, Amazon SES will automatically update your domain's DNS server with the necessary records. This can be disabled by expanding the **Advanced DKIM settings** and unchecking **Publish DNS records to Route53** in the **Easy DKIM** selection.

▼ Advanced DKIM settings

Identity type

Easy DKIM

To set up Easy DKIM, you have to modify the DNS settings for your domain.

Provide DKIM authentication token (BYODKIM)

Configure DKIM for this domain by providing your own private key.

- After you've created your domain identity with Easy DKIM, you must complete the verification process with DKIM authentication by copying the following generated CNAME records to publish to your domain's DNS provider. Detection of these records can take up to 72 hours. For more information, see [Verifying a domain identity with DKIM](#) and [Easy DKIM](#)
- Open a new browser tab and navigate to the [Lightsail console](#).
- On the Lightsail home page, choose **Domains & DNS**, then choose your domain's DNS zone.
- Add the DNS records from the Amazon SES console. For more information about editing a DNS zone in Lightsail, see the [Edit a DNS zone in Amazon Lightsail](#).

The result should look like the following example.

lightsail-demo.com
DNS zone
Global, all zones

Domains Assignments **DNS records**

DNS records

Each record in a DNS zone defines how you want to route internet traffic for your domain. For example, you can add DNS records that route traffic to your Lightsail resources, another domain, or a mail server.
[Learn more about editing DNS records](#)

[+ Add record](#)

CNAME RECORDS

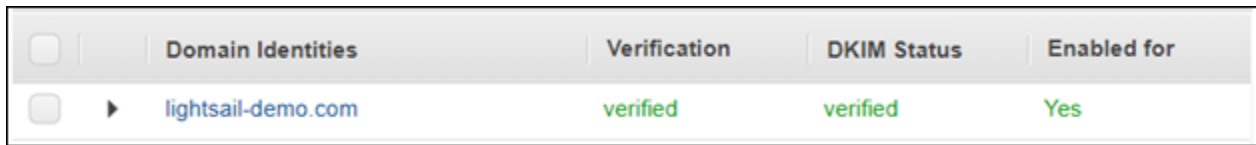
Record name	Route traffic to	
6gj4v4urninijklpgvqgiufhiiiao5f._dom...	6gj4v4urninijklpgvqgi...	
7q76h75be5hdyf7cveibg7tiy3aog54m._dom...	7q76h75be5hdyf7cveibg7...	
e5t5fevwhchlgly5puakqncvtgmneox._dom...	e5t5fevwhchlgly5puakqn...	

Note

Enter an @ symbol in the **Subdomain** text box to use the apex of your domain for an MX record. Additionally, the MX record value provided by Amazon SES is `10 inbound-smtp.us-west-2.amazonaws.com`. Enter `10` as the **Priority** and `inbound-smtp.us-west-2.amazonaws.com` as the **Maps to** domain.

8. In the [Amazon SES console](#), close the **Verify a New Domain** page.

After a few minutes, your domain listed in the Amazon SES console is labeled as verified and enabled for sending, as shown in the following example:



<input type="checkbox"/>	Domain Identities	Verification	DKIM Status	Enabled for
<input type="checkbox"/>	▶ lightsail-demo.com	verified	verified	Yes

Your SMTP service in Amazon SES is now ready to send emails from your domain.

Step 5: Verify email addresses in Amazon SES

As a new Amazon SES customer, you must verify the email addresses to which you want to send email. You do this by adding the email addresses in the Amazon SES console. For more information, see [Verifying Email Addresses in Amazon SES](#) in the Amazon SES documentation.

We recommend that you add the email addresses of your WordPress website's administrators. This lets them request password resets for their user profiles, and receive email notifications for blog posts, website updates, and other plugin messages.

Note

If you want to send email to any address without verification, then you must request to have your Amazon SES account moved out of the sandbox. For more information, see [Moving Out of the Amazon SES Sandbox](#) in the Amazon SES documentation.

To create an email address identity

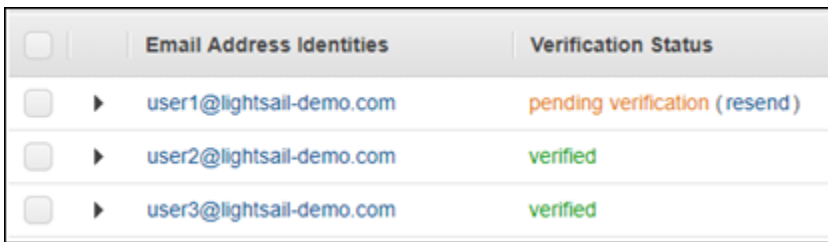
1. In the [Amazon SES console](#), from the left-navigation menu, choose **Verified identities**.
2. Choose **Create identity**.
3. Choose **Email address**. Then enter the email address that you want to verify.
4. Choose **Create identity**.

Repeat the steps 1 through 4 for every email address that you want to verify. A verification email is sent to the email address that you entered. The address is added to the list of verified email

identities with a status of "pending verification." It is marked as "verified" when the user opens the email message and completes the verification process.

To verify an email address identity

1. Check the inbox of the email address used to create your identity and look for an email from **no-reply-aws@amazon.com**.
2. Open the email and click the link to complete the verification process for the email address. After it's complete, the **Identity** status updates to **Verified**.



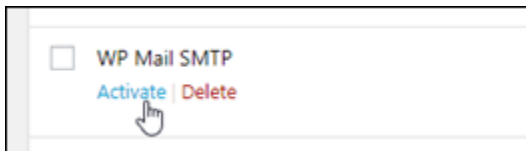
	Email Address Identities	Verification Status
<input type="checkbox"/>	▶ user1@lightsail-demo.com	pending verification (resend)
<input type="checkbox"/>	▶ user2@lightsail-demo.com	verified
<input type="checkbox"/>	▶ user3@lightsail-demo.com	verified

Step 6: Configure the WP Mail SMTP plugin on your WordPress instance

The final step is to configure the WP Mail SMTP plugin on your WordPress instance. Use the SMTP credentials that you created earlier in this guide in the Amazon SES console.

To configure the WP Mail SMTP plugin on your WordPress instance

1. Sign in to your WordPress website's dashboard as an administrator.
2. From the left-navigation menu, choose **Plugins**, then choose **Installed Plugins**.
3. Scroll down to the WP Mail SMTP plugin, then choose **Activate**. If there is a new version of the plugin, make sure to update it before continuing to the next step.



4. After the WP Mail SMTP plugin is activated, choose **Settings**. You may need to scroll back down to find the plugin.



5. In the **From Email Address** text box, enter the email address that you want emails to originate from. The email address that you enter must be confirmed in Amazon SES using the steps earlier in this guide.
6. Choose **Force From Email** to force using the email address that you enter in the **From Email Address** text box, and ignore the “from email address” value set by other plugins.
7. In the **From Name** text box, enter the name that you want emails to originate from, or leave it as is to use the name of the WordPress blog.
8. Choose **Force From Name** to force using the name that you entered in the **From Name** text box. Choosing this option ignores the “from name” value set by other plugins, and forces WordPress to use the name that you enter in the **From Name** text box.
9. In the mailer section of the page, choose **Other SMTP**.
10. Choose **Set the return-path to match the From Email** to have non-delivery receipts sent to the email address that you enter in the **From Email Address** text box.

The screenshot displays the WordPress email settings interface. It is divided into four main sections:

- From Email:** A text box contains 'user1@lightsail-demo.com'. Below it, explanatory text states: 'The email address which emails are sent from. If you using an email provider (Gmail, Yahoo, Outlook.com, etc) this should be your email address for that account. Please note that other plugins can change this, to prevent this use the setting below.' A checkbox labeled 'Force From Email' is checked, with a note: 'If checked, the From Email setting above will be used for all emails, ignoring values set by other plugins.'
- From Name:** A text box contains 'Lightsail Demo Blog'. Below it, explanatory text states: 'The name which emails are sent from.' A checkbox labeled 'Force From Name' is checked, with a note: 'If checked, the From Name setting above will be used for all emails, ignoring values set by other plugins.'
- Mailer:** A row of five mailer options is shown, each with a logo and a radio button: 'Default (none)', 'Gmail', 'Mailgun', 'SendGrid', and 'Other SMTP'. The 'Other SMTP' option is selected and highlighted with an orange border.
- Return Path:** A checkbox labeled 'Set the return-path to match the From Email' is checked. Below it, explanatory text states: 'Return Path indicates where non-delivery receipts - or bounce messages - are to be sent. If unchecked bounce messages may be lost.'

11. In the **SMTP Host** text box, enter the SMTP server name that you got earlier in this guide from the **SMTP Settings** page in the Amazon SES console.
12. Choose **TLS** in the **Encryption** section of the page to specify that the SMTP service in Amazon SES uses TLS encryption.
13. In the **SMTP Port** text box, leave the default value of **587**.
14. Switch the **Authentication** toggle to **ON**, then enter the SMTP username and password that you got earlier in this guide from the Amazon SES console.

SMTP Host

Encryption None SSL TLS
For most servers TLS is the recommended option. If your SMTP provider offers both SSL and TLS options, we recommend using TLS.

SMTP Port

Authentication ON

SMTP Username

SMTP Password
The password is stored in plain text. We highly recommend you setup your password in your WordPress configuration file for improved security; to do this add the lines below to your `wp-config.php` file.

```
define( 'WPMS_ON', true );  
define( 'WPMS_SMTP_PASS', 'your_password' );
```

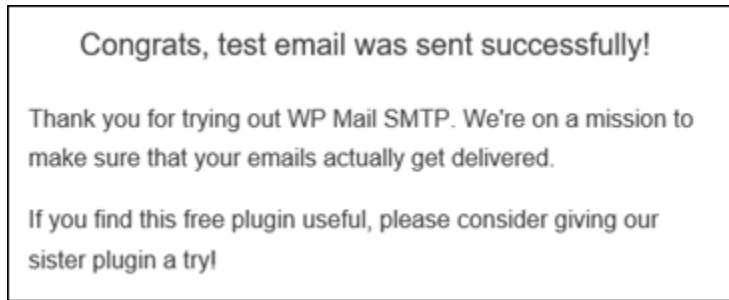
15. Choose **Save Settings**. A prompt appears confirming that the settings were successfully saved.
16. Choose the **Email Test** tab.

In the next step, you send a test email to confirm that the email service is working.

17. Enter an email address in the **Send To** text box, then choose **Send Email**. The email address that you enter must be confirmed in Amazon SES using the steps earlier in this guide.

There are two possible results that you should see.

- If you see a success confirmation, then your WordPress website is enabled for email. Confirm that the following test email arrives at the specified mailbox:



You can now choose **Lost your password?** on the sign-in page for your WordPress website's dashboard. A new password is emailed to you if the email address on your WordPress user profile is confirmed in Amazon SES.

- If you see a failure notice, confirm that the SMTP settings that you entered into the WP Mail SMTP plugin match those of the SMTP service in your Amazon SES account. Also confirm that you are using an email address that you verified in Amazon SES.

Enable HTTPS on your WordPress instance in Lightsail

Enabling Hypertext Transfer Protocol Secure (HTTPS) for your WordPress website assures visitors that your website is secure; that it's sending and receiving encrypted data. A non-secure website has an address that starts with `http`, such as `http://example.com`, while a secure website has an address that starts with `https`, such as `https://example.com`. Even if your website is primarily informational, it's still recommended that you enable HTTPS. This is because most web browsers will notify website visitors that your website is not secure if HTTPS is not enabled, and your website will rank lower in search engine results.

Tip

Lightsail offers a guided workflow that automates the installation and configuration of an SSL/TLS Let's Encrypt certificate on your WordPress instance. We highly recommend that you use the workflow instead of following the manual steps in this tutorial. For more information, see [Launch and configure a WordPress instance](#).

This guide shows you how to use the Bitnami HTTPS configuration tool (`bncert`) to enable HTTPS on your *Certified by Bitnami* WordPress instance on Amazon Lightsail. It lets you request certificates

only for the domains and subdomains that you specify when making your request. Alternately, you can use the Certbot tool, which lets you request a certificate for domains and a wildcard certificate for subdomains. A wildcard certificate works for *any* subdomains of a domain, which is beneficial if you don't know which subdomains you will use to direct traffic to your instance. However, Certbot does not automatically renew your certificate like the `bncert` tool. If you use Certbot, you have to manually renew your certificates every 90 days. For more information about using Certbot to enable HTTPS, see [Tutorial: Use Let's Encrypt SSL certificates with your WordPress instance](#).

Contents

- [Step 1: Learn about the process](#)
- [Step 2: Complete the prerequisites](#)
- [Step 3: Connect to your instance](#)
- [Step 4: Confirm the `bncert` tool is installed on your instance](#)
- [Step 5: Enable HTTPS on your WordPress instance](#)
- [Step 6: Test that your website is using HTTPS](#)

Step 1: Learn about the process

Note

In this section, you get a high-level overview of the process. The specific steps to perform this process are included in the subsequent steps of this guide.

To enable HTTPS for your WordPress website, connect to your Lightsail instance using SSH, and use the `bncert` tool to request an SSL/TLS certificate from the [Let's Encrypt](#) certificate authority. When you request the certificate, you specify your website's primary domain (`example.com`) and alternate domains (`www.example.com`, `blog.example.com`, etc.), if any. Let's Encrypt validates that you own the domains either by asking you to create TXT records in the DNS of your domains, or by verifying that those domains are already directing traffic to the public IP address of the instance from which you make the request.

After your certificate is validated, you can configure your WordPress website to automatically redirect visitors from HTTP to HTTPS (`http://example.com` redirects to `https://example.com`) so that visitors are forced to use the encrypted connection. You can also configure

your website to automatically redirect the `www` subdomain to the apex of your domain (`https://www.example.com` redirects to `https://example.com`) or vice versa (`https://example.com` redirects to `https://www.example.com`). These redirections are also configured using the `bncert` tool.

Let's Encrypt requires that you renew your certificate every 90 days to maintain HTTPS on your website. The `bncert` tool automatically renews your certificates for you, so that you can spend more time focusing on your website.

Limitations of the `bncert` tool

The `bncert` tool has the following limitations:

- It's not preinstalled on all *Certified by Bitnami* WordPress instances when they're created. WordPress instances that were created on Lightsail a while back will require that you manually install the `bncert` tool. Step 4 of this guide shows you how to confirm that the tool is installed on your instance, and how to install it if it's not.
- You can request certificates only for the domains and subdomains that you specify when making your request. This is different than the Certbot tool, which lets you request a certificate for domains and a wildcard certificate for subdomains. A wildcard certificate works for *any* subdomains of a domain, which is beneficial if you don't know which subdomains you will use to direct traffic to your instance. However, Certbot does not automatically renew your certificate like the `bncert` tool. If you use Certbot, you have to manually renew your certificates every 90 days. For more information about using Certbot to enable HTTPS, see [Tutorial: Using Let's Encrypt SSL certificates with your WordPress instance in Amazon Lightsail](#).

Step 2: Complete the prerequisites

Complete the following prerequisites if you haven't already done so:

- Create a WordPress instance in Lightsail, and configure your website on your instance. For more information, see [Get started with Linux/Unix-based instances in Amazon Lightsail](#).
- Attach a static IP to your instance. Your instance's public IP address changes if you stop and start your instance. A static IP does not change if you stop and start your instance. For more information, see [Create a static IP and attach it to an instance in Amazon Lightsail](#).
- Create a snapshot of your WordPress instance after you're done configuring it, or enable automatic snapshots. The snapshot can be used as a backup from which you can create another

instance in case something goes wrong with your original instance. For more information, see [Create a snapshot of your Linux or Unix instance](#) or [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

- Add DNS records to the DNS of your domain that directs traffic for the apex of your domain (example.com) and for its www subdomain (www.example.com) to the public IP address of your WordPress instance in Lightsail. You can complete these actions at your domain's current DNS hosting provider. Or if you transferred management of your domain's DNS to Lightsail, you can complete these actions using a DNS zone in Lightsail. To learn more, see [DNS](#).

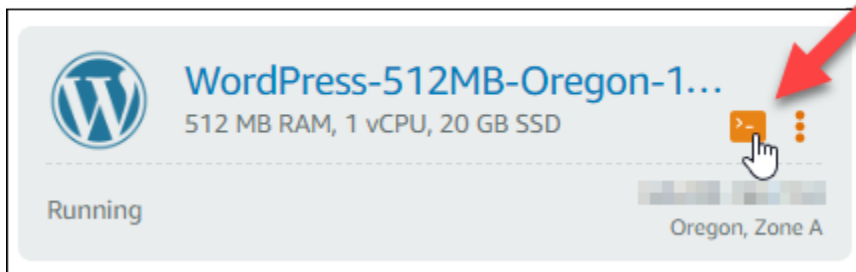
⚠ Important

Add DNS records to the DNS of all domains that you want use with your WordPress website. All of those domains should be routing traffic to the public IP address of your WordPress website. The `bncert` tool will issue certificates only for domains that are currently directing traffic to the public IP address of your WordPress instance.

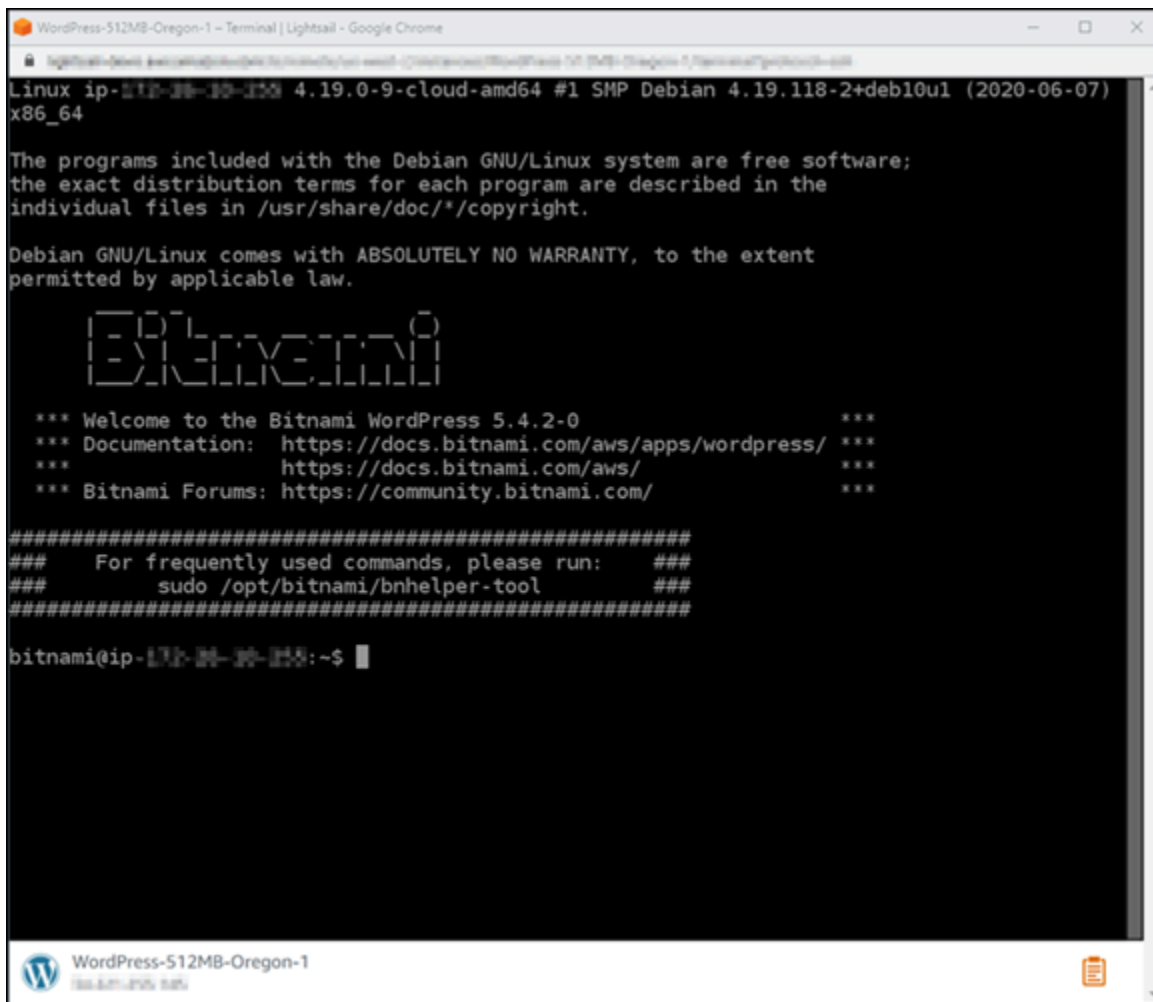
Step 3: Connect to your instance

Complete the following steps to connect to your instance using the browser-based SSH client in the Lightsail console.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the SSH quick connect icon for your WordPress instance.



The browser-based SSH client terminal window opens. You are successfully connected to your instance via SSH if you see the Bitnami logo as shown in the following example.



```
WordPress-512MB-Oregon-1 - Terminal | Lightsail - Google Chrome
Linux ip-172-31-30-150 4.19.0-9-cloud-amd64 #1 SMP Debian 4.19.118-2+deb10u1 (2020-06-07)
x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

          _   _
         | | | |
        _ |_| |_|
       |  __ | | | |
      | |  \| |
      | |__| | |
      |_____|_|_|

*** Welcome to the Bitnami WordPress 5.4.2-0 ***
*** Documentation: https://docs.bitnami.com/aws/apps/wordpress/ ***
*** https://docs.bitnami.com/aws/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***

#####
### For frequently used commands, please run: ###
### sudo /opt/bitnami/bnhelper-tool ###
#####

bitnami@ip-172-31-30-150:~$
```

Step 4: Confirm the bncert tool is installed on your instance

Complete the following steps to ensure the Bitnami HTTPS configuration tool (`bncert`) is installed on your instance. It's not preinstalled on all *Certified by Bitnami* WordPress instances when they're created. WordPress instances that were created on Lightsail a while back will require that you manually install the `bncert` tool. This procedure includes the steps to install the tool if it's not installed.

1. Enter the following command to run the `bncert` tool.

```
sudo /opt/bitnami/bncert-tool
```

- If you see `command not found` in the response as shown in the following example, then the `bncert` tool is not installed on your instance. Continue to the next step in this procedure to install the `bncert` tool on your instance.

Important

The `bncert` tool can only be used on WordPress instances that are *Certified by Bitnami*. Alternately, you can use the Certbot tool to enable HTTPS on your WordPress instance. For more information, see [Tutorial: Use Let's Encrypt SSL certificates with your WordPress instance](#).

```
bitnami@ip-172-31-15-141:~$ sudo /opt/bitnami/bncert-tool
sudo: /opt/bitnami/bncert-tool: command not found
bitnami@ip-172-31-15-141:~$
```

- If you see `Welcome to the Bitnami HTTPS configuration tool` in the response as shown in the following example, then the `bncert` tool is installed on your instance. Continue to the [Step 5: Enable HTTPS on your WordPress instance](#) section of this guide.

```
bitnami@ip-172-31-15-141:~$ sudo /opt/bitnami/bncert-tool
-----
Welcome to the Bitnami HTTPS Configuration tool.
-----
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []:
```

2. Enter the following command to download the `bncert` run file to your instance.

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/
bncert-linux-x64.run
```

3. Enter the following command to create a directory for the `bncert` run file on your instance.

```
sudo mkdir /opt/bitnami/bncert
```

4. Enter the following command to move the downloaded `bncert` run file to the new directory you created.

```
sudo mv bncert-linux-x64.run /opt/bitnami/bncert/
```

5. Enter the following command to make the `bncert` run a file that can be executed as a program.

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. Enter the following command to create a symbolic link that runs the `bncert` tool when you enter the `sudo /opt/bitnami/bncert-tool` command.

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

You are now done installing the `bncert` tool on your instance. Continue to the [Step 5: Enable HTTPS on your WordPress instance](#) section of this guide.

Step 5: Enable HTTPS on your WordPress instance

Complete the following procedure to enable HTTPS on your WordPress instance after you have confirmed that the `bncert` tool is installed on your instance.

1. Enter the following command to run the `bncert` tool.

```
sudo /opt/bitnami/bncert-tool
```

You should see a message similar to the following example.

```
bitnami@ip-172-31-1-10:~$ sudo /opt/bitnami/bncert-tool
-----
Welcome to the Bitnami HTTPS Configuration tool.
-----
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: █
```

If the `bncert` tool has been installed on your instance for a while, then you might see a message indicating that an updated version of the tool is available. Choose to download it as

shown in the following example, and then enter the `sudo /opt/bitnami/bncert-tool` command to run the `bncert` tool again.

```
bitnami@ip-10-10-10-10:~$ sudo /opt/bitnami/bncert-tool
An updated version is available. Would you like to download it? You would need to run it manually later. [Y/n]: Y
```

2. Enter your primary domain name and alternate domain names separated by a space as shown in the following example.

If your domain is not configured to route traffic to the public IP address of your instance, the `bncert` tool will ask you to make that configuration before continuing. Your domain must be routing traffic to the public IP address of the instance from which you are using the `bncert` tool to enable HTTPS on the instance. This confirms that you own the domain, and serves as the validation for your certificate.

```
-----
Welcome to the Bitnami HTTPS Configuration tool.
-----
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: example.com www.example.com
```

3. The `bncert` tool will ask you how you want your website's redirection to be configured. These are the options available:
 - **Enable HTTP to HTTPS redirection** - Specifies whether users who browse to the HTTP version of your website (i.e., `http://example.com`) are automatically redirected to the HTTPS version (i.e., `https://example.com`). We recommend enabling this option because it forces all visitors to use the encrypted connection. Type **Y** and press **Enter** to enable it.
 - **Enable non-www to www redirection** - Specifies whether users who browse to the apex of your domain (i.e., `https://example.com`) are automatically redirected to your domain's `www` subdomain (i.e., `https://www.example.com`). We recommend enabling this option. However, you may want to disable it and enable the alternate option (enable `www` to non-`www` redirection) if you have specified the apex of your domain as your preferred website address in search engine tools like Google's webmaster tools, or if your apex points directly to your IP and your `www` subdomain references your apex via a CNAME record. Type **Y** and press **Enter** to enable it.
 - **Enable www to non-www redirection** - Specifies whether users who browse to your domain's `www` subdomain (i.e., `https://www.example.com`) are automatically redirected

to the apex of your domain (i.e., `https://example.com`). We recommend disabling this, if you enabled non-`www` redirection to `www`. Type `N` and press **Enter** to disable it.

Your selections should look like the following example.

```
Enable/disable redirections

Please select the redirections you wish to enable or disable on your Bitnami
installation.

Enable HTTP to HTTPS redirection [Y/n]: Y

Enable non-www to www redirection [Y/n]: Y

Enable www to non-www redirection [y/N]: N
```

4. The changes that are going to be made are listed. Type `Y` and press **Enter** to confirm and continue.

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
www.example.com)
7. Start web server once all changes have been performed

Do you agree to these changes? [Y/n]: Y
```

5. Enter your email address to associate with your Let's Encrypt certificate and press **Enter**.

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []: █
```

6. Review the Let's Encrypt Subscriber Agreement. Type Y and press **Enter** to accept the agreement and continue.

```
The Let's Encrypt Subscriber Agreement can be found at:
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf
Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]: █
```

The actions are performed to enable HTTPS on your instance, including requesting the certificate and configuring the redirections you specified.

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.

█
```

Your certificate is successfully issued and validated, and the redirections are successfully configured on your instance if you see a message similar to the following example.

```
Success

The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.
The configuration report is shown below.

Backup files:
* /opt/bitnami/apache2/conf/httpd.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami.conf.back.202005290035

Find more details in the log file:
/tmp/bncert-202005290035.log

If you find any issues, please check Bitnami Support forums at:
https://community.bitnami.com

Press [Enter] to continue:█
```


The `bncert` tool will perform an automatic renewal of your certificate every 80 days before it expires. Repeat the above steps if you wish to use additional domains and subdomains with your instance, and you want to enable HTTPS for those domains.

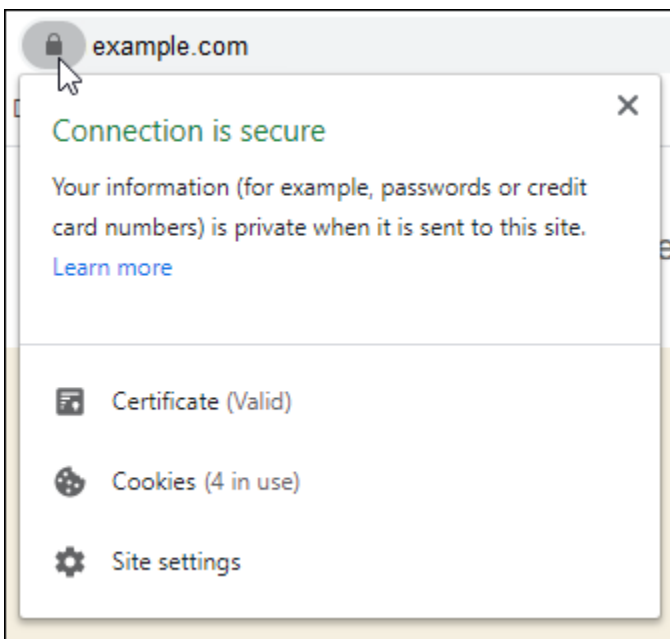
You are now done enabling HTTPS on your WordPress instance. Continue to the [Step 6: Test that your website is using HTTPS](#) section of this guide.

Step 6: Test that your website is using HTTPS

After you enable HTTPS on your WordPress instance, you should confirm that your website is using HTTPS by browsing to all of the domains that you specified when using the `bncert` tool. When you visit each domain, you should see that they use a secure connection as shown in the following example.

Note

You might have to refresh, and clear your browser's cache to see the change.



You might also notice that the non-`www` address redirects to the `www` subdomain of your domain, or vice versa depending on the option you selected when running the `bncert` tool.

Migrate an existing WordPress blog to Amazon Lightsail

Looking to change your WordPress hosting provider? Amazon Lightsail is the easiest way to run a WordPress site on AWS.

You can choose one of our pricing plans (starting at \$3.50 USD per month) and have full control over your WordPress installation, including plugins, themes, and more.

Creating a Lightsail WordPress instance only takes a few minutes. Follow this tutorial to back up your existing WordPress blog and import it to a new instance running in Lightsail.

Here's a quick overview of the process:



Continue reading to get started.

Prerequisites

Before you begin, you'll need the following:

1. You'll need to an AWS account. [Sign up for AWS](#), or [sign in to AWS](#) if you already have an account.
2. Make sure your account is set up to use Lightsail. If it has been a while since you created your account, or if you haven't provided a credit card yet, you may need to log in to the AWS Management Console and update your account first.

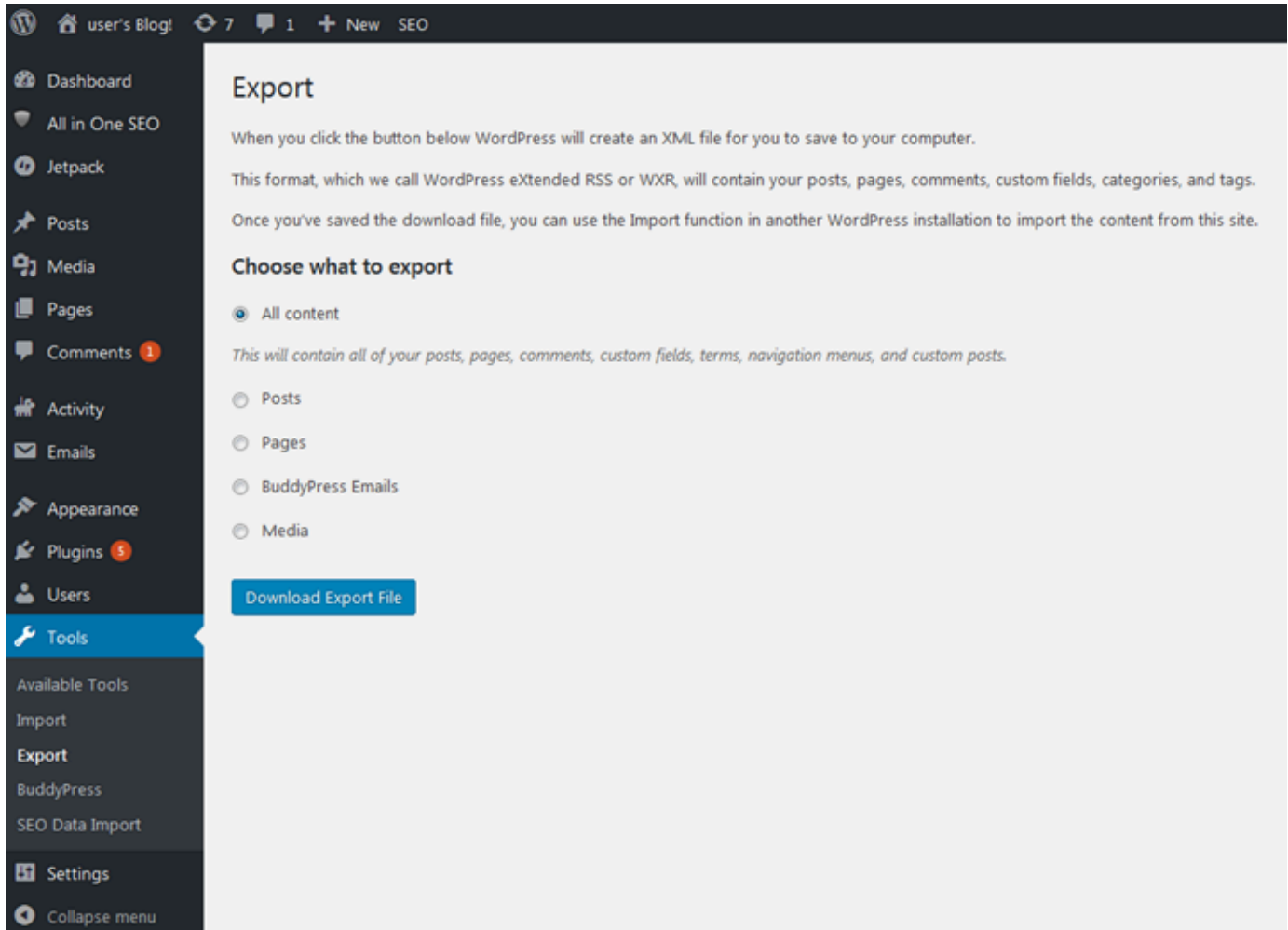
Step 1: Back up your existing WordPress blog

You can use WordPress to back up your existing blog. You'll just need to be able to log into the WordPress admin console and manage your blog.

1. Navigate to your blog, and then choose **Manage**.

If the **Manage** banner is not shown, you can reach the sign in page by browsing to `http://<PublicIP>/wp-login.php`. Replace `<PublicIP>` with the public IP address of your instance.

2. Enter your user name and password to log into the WordPress admin console.
3. On the WordPress **Dashboard**, choose **Tools**, and then choose **Export**.
4. On the **Export** page, choose **All content** to export everything as an XML file.



5. Choose **Download export file** to download your old blog as an XML file.

Save the XML file in a location that's easy to find. You'll need it in Step 4.

Step 2: Create a new WordPress instance in Lightsail

You can create a new WordPress instance in Lightsail in just a few minutes. Here's how:











1. Go to the [Lightsail home page](#) and log in.
2. Choose **Create instance**.
3. Select the AWS Region where you'd like to create your blog.

You can choose the default Availability Zone or change that once you select an AWS Region.

4. Select **WordPress**.

Pick your instance image [?](#)

Apps + OS OS Only

 WordPress 4.7.3	 LAMP Stack 5.6.30	 Node.js 7.7.1	 Joomla 3.6.5
 Magento 2.1.5	 MEAN 3.4.2	 Drupal 8.2.7	 GitLab CE 8.16.4
 Redmine 3.3.2	 Nginx 1.10.3		

WordPress 4.7.3

WordPress powered by Bitnami and sold by BitRock Inc. is a pre-configured, ready to run image for running WordPress on Amazon EC2. WordPress is one of the world's most popular web publishing platforms for building blogs and websites. It can be customized via a wide selection of themes, extensions and plug-ins.

Learn more about WordPress on the [AWS Marketplace](#) .

By using this image, you agree to the provider's [End User License Agreement](#) .

5. Choose your instance plan (or *bundle*).

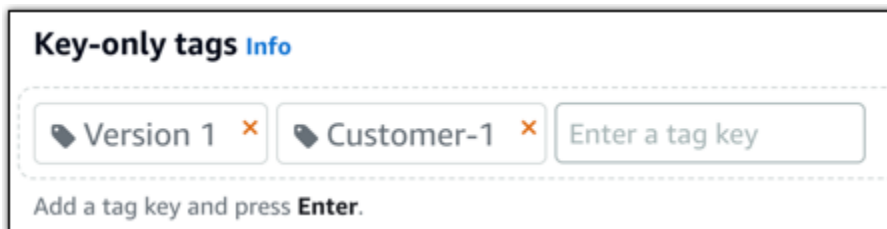
You can upgrade your Lightsail plan later if needed. For more information, see [Create an instance from a snapshot in Lightsail](#).

6. Enter a name for your instance.

Resource names:

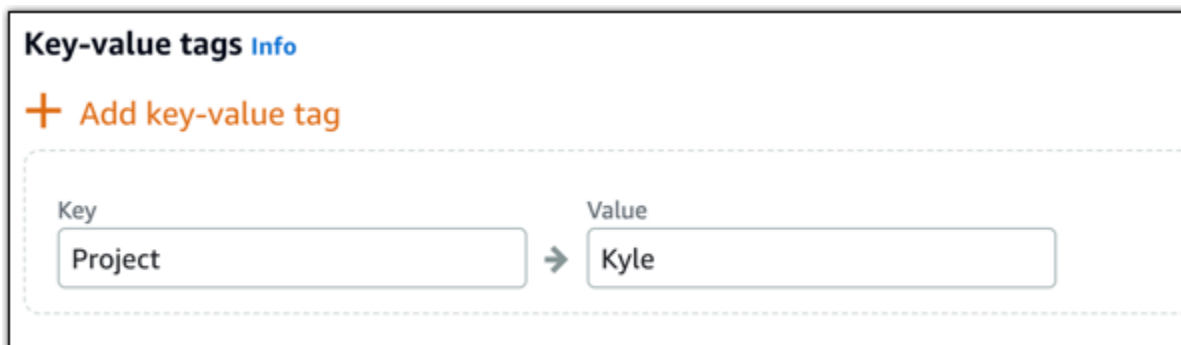
- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2–255 characters.

- Must start and end with an alphanumeric character.
 - Can include alphanumeric characters, periods, dashes, and underscores.
7. Choose one of the following options to add tags to your instance:
- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Note

For more information about key-only and key-value tags, see [Tags](#).

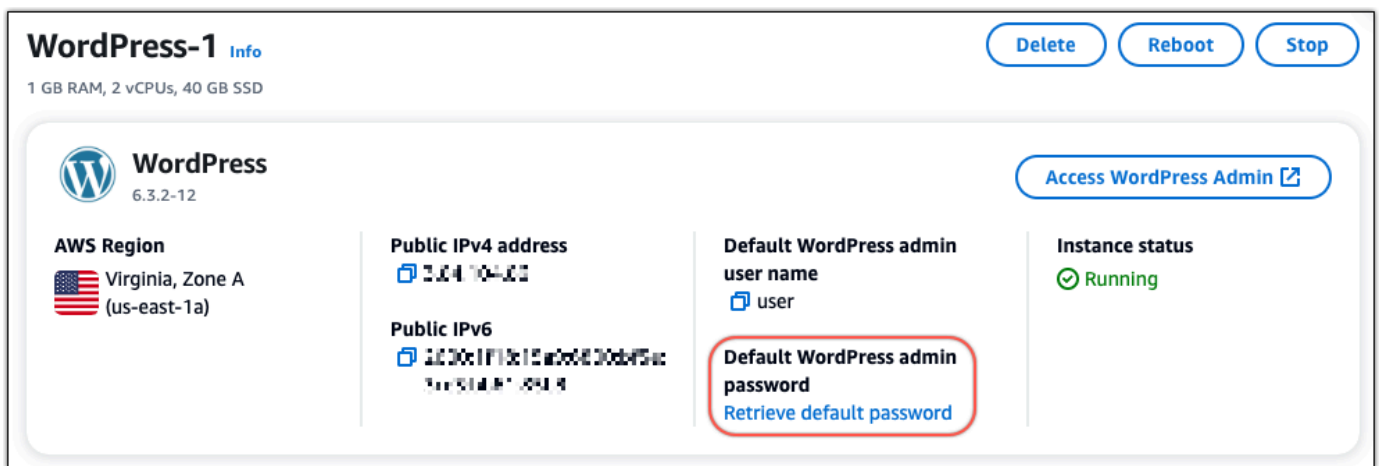
8. Choose **Create instance**.

Step 3: Log into your new Lightsail WordPress blog

Now that you have a new blog in Lightsail, you'll need to access the WordPress Dashboard to import your old blog data. The default password to sign in to the administration dashboard of your WordPress website is stored on the instance. Complete the following steps to get the password.

To get the default password for the WordPress administrator

1. Open the instance management page for your WordPress instance.
2. On the **WordPress** panel, choose **Retrieve default password**. This expands **Access default password** at the bottom of the page.



3. Choose **Launch CloudShell**. This opens a panel at the bottom of the page.
4. Choose **Copy** and then paste the contents into the CloudShell window. You can either put your cursor at the CloudShell prompt and press Ctrl+V, or you can right-click to open the menu and then choose **Paste**.
5. Make a note of the password displayed in the CloudShell window. You need this to sign in to the administration dashboard of your WordPress website.

```
[cloudshell-user@ip-10-114-41-117 ~]$ AWS_REGION=us-east-1 ~/lightsail_connect WordPress-1 cat bitnami_application_password
JKzh8wB5FAR!
```

Now that you have the password for the administration dashboard of your WordPress website, you can sign in. In the administration dashboard, you can change your user password, install plugins, change the theme of your website, and more.

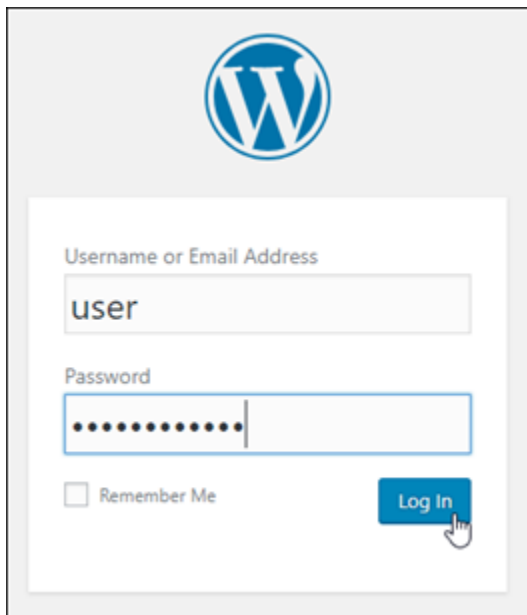
Complete the following steps to sign in to the administration dashboard of your WordPress website.

To sign in to the administration dashboard

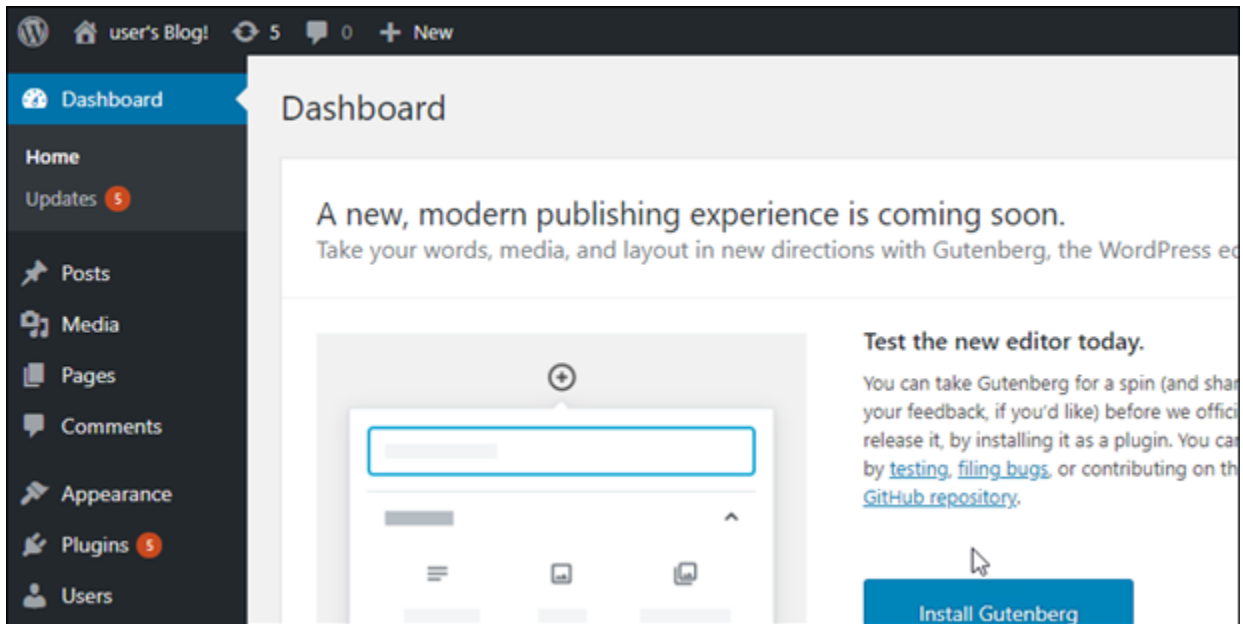
1. Open the instance management page for your WordPress instance.
2. On the **WordPress** panel, choose **Access WordPress Admin**.
3. On the **Access your WordPress Admin Dashboard** panel, under **Use public IP address**, choose the link with this format:

`http://public-ipv4-address./wp-admin`

4. For **Username or Email Address**, enter **user**.
5. For **Password**, enter the password obtained in the previous step.
6. Choose **Log in**.

A screenshot of the WordPress login interface. At the top center is the WordPress logo (a blue 'W' inside a circle). Below it is a white login form with a light gray border. The form contains two input fields: 'Username or Email Address' with the text 'user' entered, and 'Password' with a series of black dots. Below the password field is a checkbox labeled 'Remember Me' and a blue 'Log In' button with a mouse cursor hovering over it.

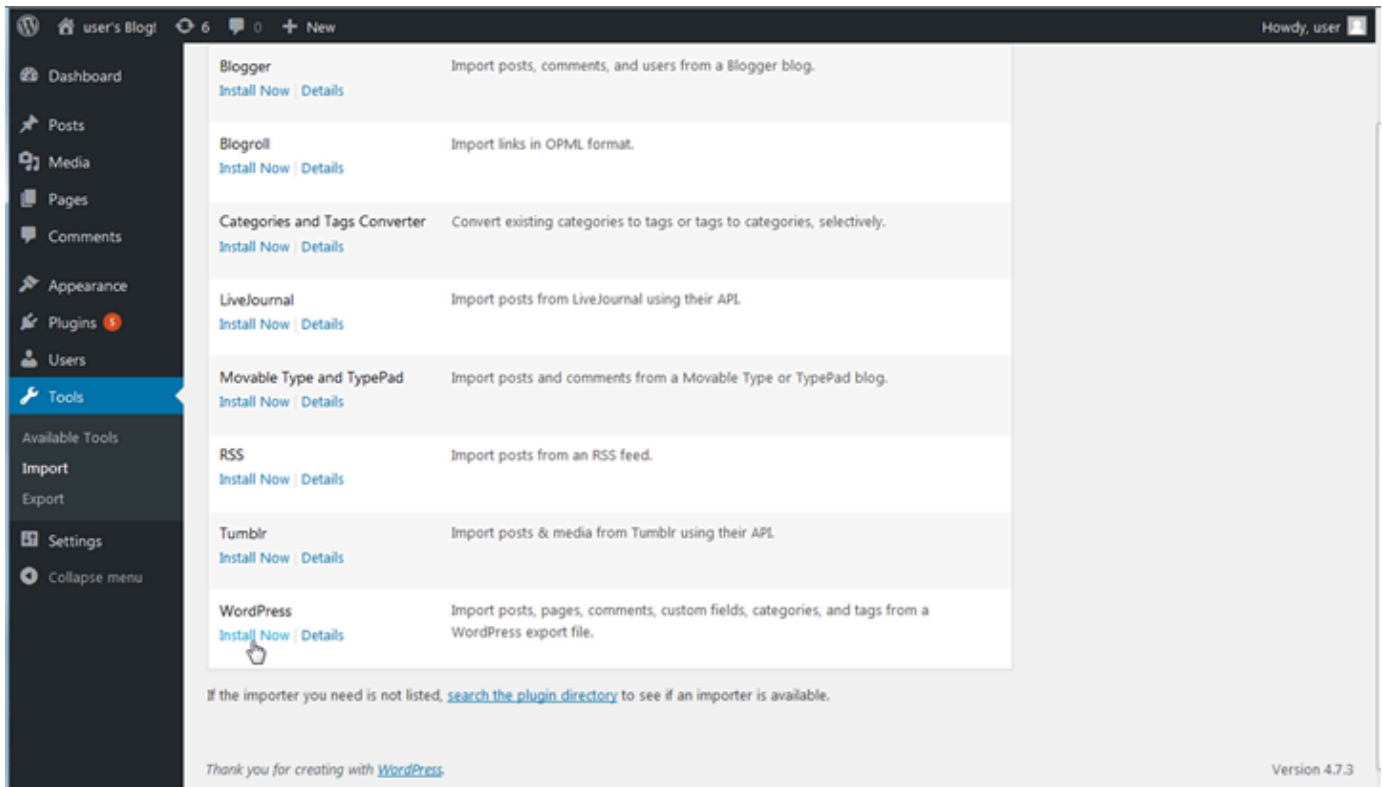
You are now signed in to the administration dashboard of your WordPress website where you can perform administrative actions. For more information about administering your WordPress website, see the [WordPress Codex](#) in the WordPress documentation.



Step 4: Import your XML file into your new Lightsail blog

Once you have successfully logged into the WordPress Dashboard on your new Lightsail instance, follow these steps to import the XML file into your new Lightsail blog.

1. From the WordPress **Dashboard** on your new Lightsail instance, choose **Tools**.
2. Choose **Import**, and then choose **Install Now** to install the WordPress import tool.



3. Once the tool is done installing, choose **Run Importer** to run the import tool.
4. On the **Import WordPress** page, choose **Browse**.
5. Find the XML file you saved in *Step 1: Back up your existing WordPress blog*, and then choose **Open**.
6. Choose **Upload file and import**.

Accept the rest of the defaults, and then choose **Submit**.

Next steps

You can verify that everything worked by choosing your blog (next to the Home icon), and then choosing **Visit Site** from the WordPress dashboard. You can also type the IP address into a browser and view the blog.

Here are some next steps:

- Migrate your DNS so that your domain name servers point to the new version of your blog.
- Customize your new blog's appearance and/or install some WordPress plugins.
- [Enable HTTPS support with SSL certificates](#)

WordPress Multisite tutorials for Amazon Lightsail

WordPress Multisite enables administrators to host and manage multiple websites from the same WordPress instance. Use the following tutorials to learn how to work with WordPress Multisite in Lightsail.

Topics

- [Add blogs as domains to your WordPress Multisite instance in Lightsail](#)
- [Add blogs as subdomains to your WordPress Multisite instance in Lightsail](#)
- [Define the primary domain for your WordPress Multisite instance in Lightsail](#)

Add blogs as domains to your WordPress Multisite instance in Lightsail

A WordPress Multisite instance in Amazon Lightsail is designed to use multiple domains, or subdomains, for each blog site that you create within that instance. In this guide, we'll show you how to add a blog site using a different domain than your main blog's primary domain on your WordPress Multisite instance. For example, if your main blog's primary domain is `example.com`, you can create new blog sites that use the `another-example.com` and `third-example.com` domains on the same instance.

Note

You can also add sites using subdomains to your WordPress Multisite instance. For more information, see [Add blogs as subdomains to your WordPress Multisite instance](#).

Prerequisites

Complete the following prerequisites in the order shown:

1. Create a WordPress Multisite instance in Lightsail. For more information, see [Create an instance](#).
2. Create a static IP and attach it to your WordPress Multisite instance in Lightsail. For more information, see [Create a static IP and attach it to an instance](#).
3. Add your domain to Lightsail by creating a DNS zone, then point it to the static IP that you attached to your WordPress Multisite instance. For more information, see [Create a DNS zone to manage your domain's DNS records](#).

4. Define the primary domain for your WordPress Multisite instance. For more information, see [Define the primary domain for your WordPress Multisite instance](#).

Add a blog as a domain to your WordPress Multisite instance

Complete these steps to create a blog site on your WordPress Multisite instance that uses a domain which is different than your main blog's primary domain.

Important

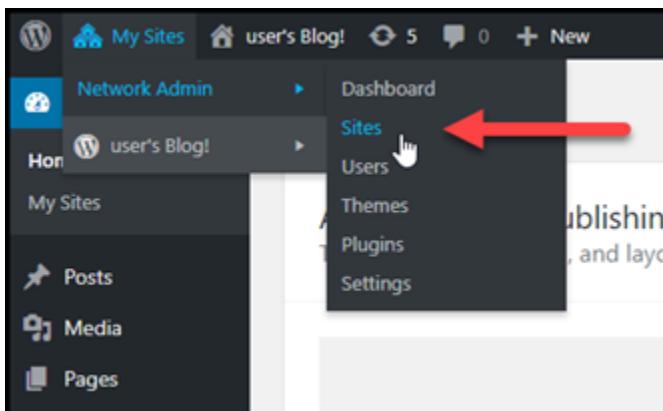
You must complete step 4 listed in the prerequisites section of this guide before following these steps.

1. Sign in to the administration dashboard of your WordPress Multisite instance.

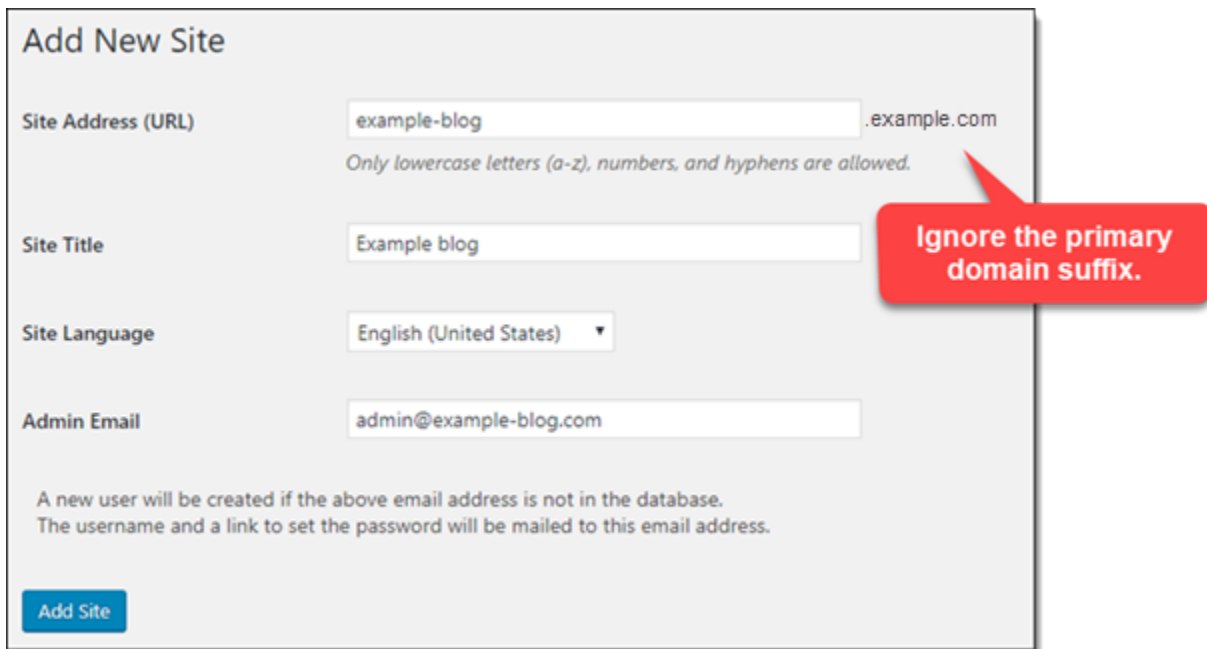
Note

For more information, see [Get the application user name and password for your Bitnami instance](#).

2. Choose **My Sites**, then **Network Admin**, and **Sites** in the top navigation pane.



3. Choose **Add New** to add a new blog site.
4. Enter a site address into the **Site Address (URL)** text box. This is domain that will be used for the new blog site. For example, if your new blog site will use `example-blog.com` as the domain, then enter `example-blog` into the **Site Address (URL)** text box. Ignore the primary domain suffix displayed on the page.



Add New Site

Site Address (URL) .example.com
Only lowercase letters (a-z), numbers, and hyphens are allowed.

Site Title

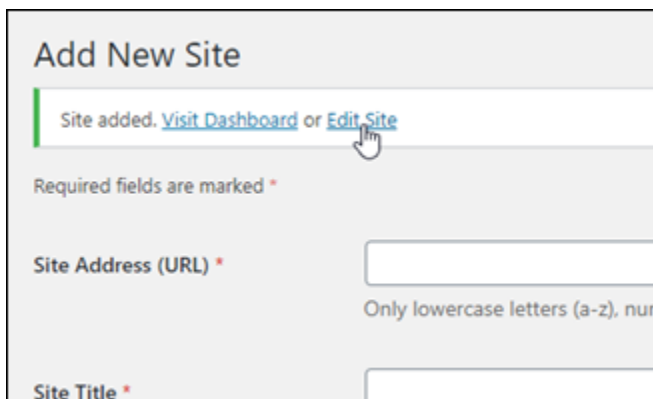
Site Language

Admin Email

A new user will be created if the above email address is not in the database.
The username and a link to set the password will be mailed to this email address.

[Add Site](#)

5. Enter a site title, select a site language, and enter an admin email.
6. Choose **Add Site**.
7. Choose **Edit Site** in the confirmation banner that appears on the page. This will redirect you to edit the details of the site that you recently created.



Add New Site

Site added. [Visit Dashboard](#) or [Edit Site](#)

Required fields are marked *

Site Address (URL) *
Only lowercase letters (a-z), num

Site Title *

8. In the **Edit Site** page, change the subdomain that is listed in the **Site Address (URL)** text box to the apex domain that you want to use. In this example, we specified `http://example-blog.com`.

Edit Site: Example Blog

[Visit](#) | [Dashboard](#)

Info Users Themes Settings

Site Address (URL)

Registered

Last Updated

Attributes

- Public
- Archived
- Spam
- Deleted
- Mature

9. Choose **Save Changes**.

At this point, the new blog site has been created in your WordPress Multisite instance, but the domain is not yet configured to route to the new blog site. Continue to the next step to add an address record (A record) to your domain's DNS zone.

Sites Screen Options ▾ Help ▾

All (2) | Public (2)

Bulk actions ▾ 2 items

<input type="checkbox"/>	URL	Last Updated	Registered	Users
<input type="checkbox"/>	example.com — Main	Never	2020/12/10	1
<input type="checkbox"/>	example-blog.com	2021/01/25	2021/01/25	1

URL Last Updated Registered Users

Bulk actions ▾ 2 items

Add an address record (A record) to your domain's DNS zone

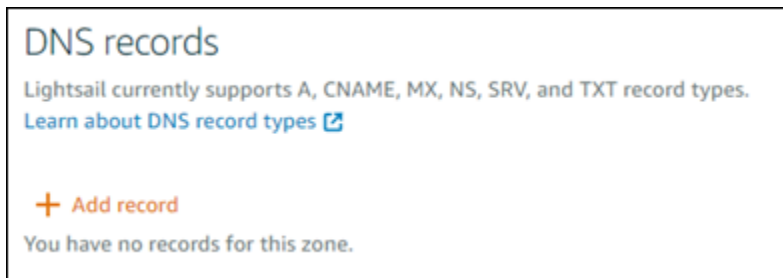
Complete these steps to point the domain for your new blog site to your WordPress Multisite instance. You must perform these steps for every blog site that you create on your WordPress Multisite instance.

For demonstration purposes, we'll use the Lightsail DNS zone. However, the steps may be similar for other DNS zones typically hosted by domain registrars.

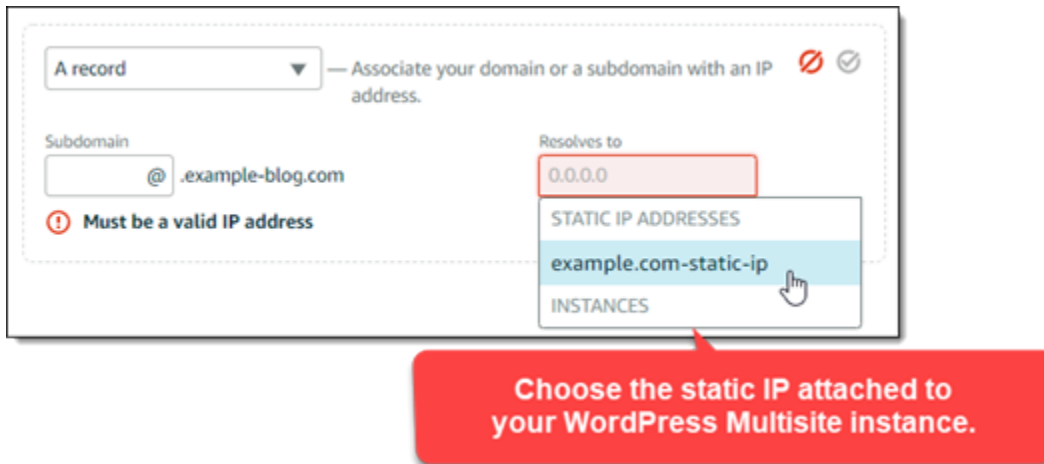
Important

You can create a maximum of six DNS zones in the Lightsail console. If you need more DNS zones, we recommend using Amazon Route 53 to manage your domain's DNS records. For more information, see [Make Amazon Route 53 the DNS service for an existing domain](#).

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Domains & DNS** tab.
3. Under the **DNS zones** section of the page, choose the DNS zone for your new blog site's domain.
4. In the DNS zone editor, choose the **DNS records** tab. Then, choose **Add record**.



5. Choose **A record** in the record type drop-down menu.
6. In the **Record name** text box, enter an "at" (@) symbol to create a record for the root of the domain.
7. In the **Resolves to** text box, choose the static IP address attached to your WordPress Multisite instance.



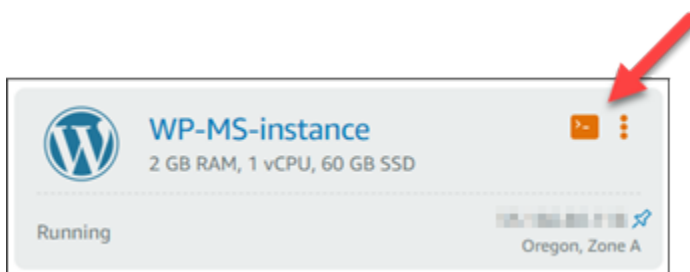
8. Choose the Save icon.

After the change propagates through the internet's DNS, the domain will route traffic to the new blog site on your WordPress Multisite instance.

Enable cookie support to allow sign in for blog sites

When you add blog sites as domains to your WordPress Multisite instance, you must also update the WordPress configuration (`wp-config`) file on your instance to enable cookie support. If you don't enable cookie support, then users might experience a "Error: Cookies are blocked or not supported" error when trying to sign in to the WordPress administration dashboard of their blog sites.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the SSH quick connect icon for your WordPress Multisite instance.



3. After your Lightsail browser-based SSH session is connected, enter the following command to open and edit the `wp-config.php` file of your instance using Vim:

```
sudo vim /opt/bitnami/wordpress/wp-config.php
```

Note

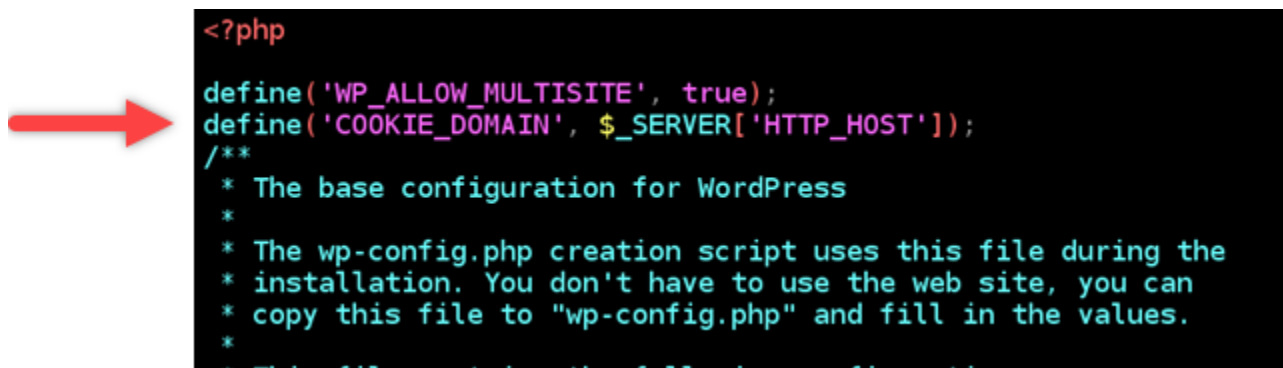
If this command fails, you might be using an older version of the WordPress Multisite instance. Try running the following command instead.

```
sudo vim /opt/bitnami/wordpress/wp-config.php
```

4. Press **I** to enter insert mode in Vim.
5. Add the following line of text below the `define('WP_ALLOW_MULTISITE', true);` line of text.

```
define('COOKIE_DOMAIN', $_SERVER['HTTP_HOST']);
```

The file will look like the following when done:



```
<?php
define('WP_ALLOW_MULTISITE', true);
define('COOKIE_DOMAIN', $_SERVER['HTTP_HOST']);
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configuration:
 *
 * - MySQL database name
 * - MySQL database user name
 * - MySQL database password
 * - WordPress table prefix
 *
 * For more information on this file, see the WordPress Codex page on
 * wp-config.php: https://codex.wordpress.org/Editing_wp-config.php
 */

```

6. Press the **Esc** key to exit insert mode in Vim, then type `:wq!` and press **Enter** to save your edits (write) and quit Vim.
7. Enter the following command to restart the underlying services of the WordPress instance.

```
sudo /opt/bitnami/ctlscript.sh restart
```

Cookies should now be enabled on your WordPress multisite instance, and users who are trying to sign in to their blog sites will not encounter the "Error: Cookies are blocked or not supported" error.

Next steps

After you add blogs as domains to your WordPress Multisite instance, we recommend that you get familiar with WordPress Multisite administration. For more information see [Multisite Network Administration](#) in the WordPress documentation.

Add blogs as subdomains to your WordPress Multisite instance in Lightsail

A WordPress Multisite instance in Amazon Lightsail is designed to use multiple domains, or subdomains, for each blog site that you create within that instance. In this guide, we'll show you how to add a blog site as a subdomain of your WordPress Multisite instance. For example, if your main blog's primary domain is `example.com`, you can create new blog sites that use the `earth.example.com` and `moon.example.com` subdomains on the same instance.

Note

You can also add sites using domains to your WordPress Multisite instance. For more information, see [Add blogs as domains to your WordPress Multisite instance](#).

Prerequisites

Complete the following prerequisites in the order shown:

1. Create a WordPress Multisite instance. For more information, see [Create an instance](#).
2. Create a static IP and attach it to your WordPress Multisite instance. For more information, see [Create a static IP and attach it to an instance](#).
3. Add your domain to Lightsail by creating a DNS zone, then point it to the static IP that you attached to your WordPress Multisite instance. For more information, see [Create a DNS zone to manage your domain's DNS records](#).
4. Define the primary domain for your WordPress Multisite instance. For more information, see [Define the primary domain for your WordPress Multisite instance](#).

Add a blog as a subdomain to your WordPress Multisite instance

Complete these steps to create new blogs on your WordPress Multisite instance that use a subdomain of your main blog's primary domain.

⚠ Important

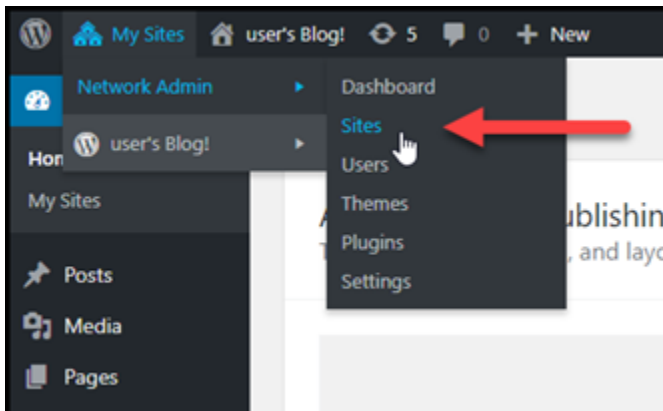
You must complete step 4 listed in the prerequisites section of this guide before following these steps.

1. Sign in to the administration dashboard of your WordPress Multisite instance.

📘 Note

For more information, see [Get the application user name and password for your Bitnami instance](#).

2. Choose **My Sites**, then **Network Admin**, and **Sites** in the top navigation pane.



3. Choose **Add New** to add a new blog site.
4. Enter a site address, which is the subdomain that will be used for the new blog site.

Add New Site

Site Address (URL) .example.com
Only lowercase letters (a-z), numbers, and hyphens are allowed.

Site Title

Site Language

Admin Email

A new user will be created if the above email address is not in the database.
The username and a link to set the password will be mailed to this email address.

5. Enter a site title, select a site language, and enter an admin email.
6. Choose **Add Site**.

At this point, the new blog site has been created in your WordPress Multisite instance, but the subdomain is not yet configured to route to the new blog site. Continue to the next step to add an address record (A record) to your domain's DNS zone.

Sites

Bulk Actions 3 items

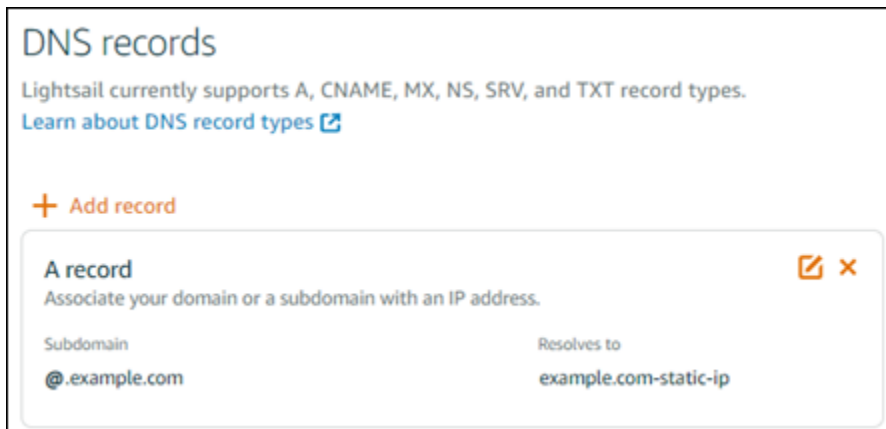
<input type="checkbox"/>	URL	Last Updated	Registered	Users
<input type="checkbox"/>	example.com	Never	2018/08/15	1
<input type="checkbox"/>	earth.example.com	2018/10/22	2018/10/22	1
<input type="checkbox"/>	moon.example.com	2018/10/22	2018/10/22	1
<input type="checkbox"/>	URL	Last Updated	Registered	Users

Add an address record (A record) to your domain's DNS zone

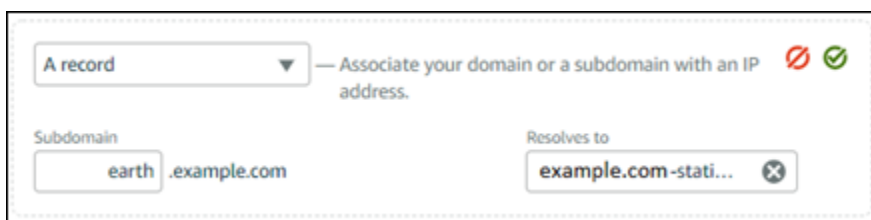
Complete these steps to point the subdomain for your new blog site to your WordPress Multisite instance. You must perform these steps for every blog site that you create on your WordPress Multisite instance.

For demonstration purposes, we'll use the Lightsail DNS zone. However, the steps may be similar for other DNS zones typically hosted by domain registrars.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Domains & DNS** tab.
3. Under the **DNS zones** section of the page, choose the DNS zone for the domain that you defined as the primary domain for your WordPress Multisite instance.
4. In the DNS zone editor, choose the **DNS records** tab. Then, choose **Add record**.



5. Choose **A record** in the record type drop-down menu.
6. In the **Record name** text box, enter the subdomain specified as the site address when creating the new blog site on your WordPress Multisite instance.
7. In the **Resolves to** text box, choose the static IP address attached to your WordPress Multisite instance.



8. Choose the Save icon.

That is all you need to do. After the change propagates through the internet's DNS, the domain will redirect to the new blog site on your WordPress Multisite instance.

Next steps

After you add blogs as subdomains to your WordPress Multisite instance, we recommend that you get familiar with WordPress Multisite administration. For more information see [Multisite Network Administration](#) in the WordPress documentation.

Define the primary domain for your WordPress Multisite instance in Lightsail

A WordPress Multisite instance in Amazon Lightsail is designed to use multiple domains, or subdomains, for each blog site that you create within that instance. Because of this, you must define the primary domain to use for the main blog of your WordPress Multisite instance.

Prerequisites

Complete the following prerequisites in the order shown:

1. Create a WordPress Multisite instance in Lightsail. For more information, see [Create an instance](#).
2. Create a static IP and attach it to your WordPress Multisite instance in Lightsail. For more information, see [Create a static IP and attach it to an instance](#).

Important

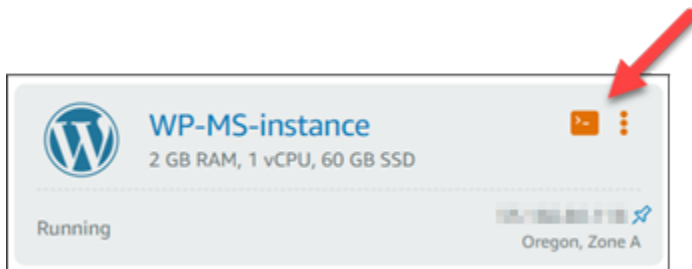
You must reboot your WordPress Multisite instance after you attach a static IP to it. This will allow the instance to recognize the new static IP associated to it.

3. Add your domain to Lightsail by creating a DNS zone, then point it to the static IP that you attached to your WordPress Multisite instance. For more information, see [Create a DNS zone to manage your domain's DNS records](#).
4. Allow time for the DNS changes to propagate through the internet's DNS. Then, you can continue to the [Define the primary domain for your WordPress Multisite instance](#) section of this guide.

Define the primary domain for your WordPress Multisite instance

Complete these steps to ensure that your domain, such as `example.com`, redirects to the main blog of your WordPress Multisite instance.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the SSH quick connect icon for your WordPress Multisite instance.



3. Enter the following command to define the primary domain name for your WordPress Multisite instance. Be sure to replace `<domain>` with the correct domain name for your WordPress Multisite.

```
sudo /opt/bitnami/configure_app_domain --domain <domain>
```

Example:

```
sudo /opt/bitnami/configure_app_domain --domain example.com
```

Note

If this command fails, you might be using an older version of the WordPress Multisite instance. Try running the following commands instead, and be sure to replace `<domain>` with the correct domain name for your WordPress Multisite.

```
cd /opt/bitnami/apps/wordpress  
sudo ./bnconfig --machine_hostname <domain>
```

After running that command, enter the following command to keep the `bnconfig` tool from automatically running every time the server restarts.

```
sudo mv bnconfig bnconfig.disabled
```

At this point, browsing to the domain that you defined should redirect you to the main blog of your WordPress Multisite instance.

Next steps

Complete the next steps after you have defined the primary domain for your WordPress Multisite instance:

- [Add blogs as subdomains to your WordPress Multisite instance](#)
- [Add blogs as domains to your WordPress Multisite instance](#)

Let's Encrypt tutorials for Amazon Lightsail

Let's Encrypt issues free SSL/TLS certificates, enabling secure and encrypted communication for websites, applications, and online services. Use the following tutorials to learn how to work with Let's Encrypt in Lightsail.

Topics

- [Tutorial: Use Let's Encrypt SSL certificates with your Lightsail LAMP instance](#)
- [Tutorial: Use Let's Encrypt SSL certificates with your Lightsail Nginx instance](#)
- [Tutorial: Use Let's Encrypt SSL certificates with your Lightsail WordPress instance](#)

Tutorial: Use Let's Encrypt SSL certificates with your Lightsail LAMP instance

Amazon Lightsail makes it easy to secure your websites and applications with SSL/TLS using Lightsail load balancers. However, using a Lightsail load balancer might not generally be the right choice. Perhaps your site doesn't need the scalability or fault tolerance load balancers provide, or maybe you're optimizing for cost.

In the latter case, you might consider using Let's Encrypt to obtain a free SSL certificate. If so, that's no problem. You can integrate those certificates with Lightsail instances. This tutorial shows you

how to request a Let's Encrypt wildcard certificate using Certbot, and integrate it with your LAMP instance.

Important

- The Linux distribution used by Bitnami instances changed from Ubuntu to Debian in July, 2020. Because of this change, some of the steps in this tutorial will differ depending on the Linux distribution of your instance. All Bitnami blueprint instances created after the change use the Debian Linux distribution. Instances created before the change will continue to use the Ubuntu Linux distribution. To check the distribution of your instance, run the `uname -a` command. The response will show either Ubuntu or Debian as your instance's Linux distribution.
- Bitnami is in the process of modifying the file structure for many of their stacks. The file paths in this tutorial may change depending on whether your Bitnami stack uses native Linux system packages (Approach A), or if it is a self-contained installation (Approach B). To identify your Bitnami installation type and which approach to follow, run the following command:

```
test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach A: Using system packages." || echo "Approach B: Self-contained installation."
```

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Install Certbot on your instance](#)
- [Step 3: Request a Let's Encrypt SSL wildcard certificate](#)
- [Step 4: Add TXT records to your domain's DNS zone](#)
- [Step 5: Confirm that the TXT records have propagated](#)
- [Step 6: Complete the Let's Encrypt SSL certificate request](#)
- [Step 7: Create links to the Let's Encrypt certificate files in the Apache server directory](#)
- [Step 8: Configure HTTP to HTTPS redirection for your web application](#)
- [Step 9: Renew the Let's Encrypt certificates every 90 days](#)

Step 1: Complete the prerequisites

Complete the following prerequisites if you haven't already done so:

- Create a LAMP instance in Lightsail. To learn more, see [Create an instance](#).
- Register a domain name, and get administrative access to edit its DNS records. To learn more, see [Amazon Lightsail DNS](#).

Note

We recommend that you manage your domain's DNS records using a Lightsail DNS zone. To learn more, see [Creating a DNS zone to manage your domain's DNS records](#).

- Use the browser-based SSH terminal in the Lightsail console to perform the steps in this tutorial. However, you can also use your own SSH client, such as PuTTY. To learn more about configuring PuTTY, see [Download and set up PuTTY to connect using SSH](#).

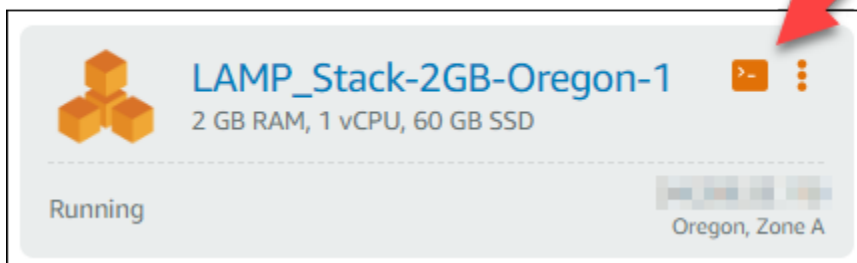
After you've completed the prerequisites, continue to the [next section](#) of this tutorial.

Step 2: Install Certbot on your instance

Certbot is a client used to request a certificate from Let's Encrypt and deploy it to a web server. Let's Encrypt uses the ACME protocol to issue certificates, and Certbot is an ACME-enabled client that interacts with Let's Encrypt.

To install Certbot on your Lightsail instance

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the SSH quick connect icon for the instance that you want to connect to.



6. Enter the following command to update apt to include the new repository:

```
sudo apt-get update -y
```

7. Enter the following command to install Certbot:

```
sudo apt-get install certbot -y
```

Certbot is now installed on your Lightsail instance.

8. Keep the browser-based SSH terminal window open—you return to it later in this tutorial. Continue to the [next section](#) of this tutorial.

Step 3: Request a Let's Encrypt SSL wildcard certificate

Begin the process of requesting a certificate from Let's Encrypt. Using Certbot, request a wildcard certificate, which lets you use a single certificate for a domain and its subdomains. For example, a single wildcard certificate works for the `example.com` top-level domain, and the `blog.example.com`, and `stuff.example.com` subdomains.

To request a Let's Encrypt SSL wildcard certificate

1. In the same browser-based SSH terminal window used in [step 2](#) of this tutorial, enter the following commands to set an environment variable for your domain. You can now more efficiently copy and paste commands to obtain the certificate.

```
DOMAIN=Domain
```

```
WILDCARD=*.$DOMAIN
```

In the command, replace *Domain* with your registered domain name.

Example:

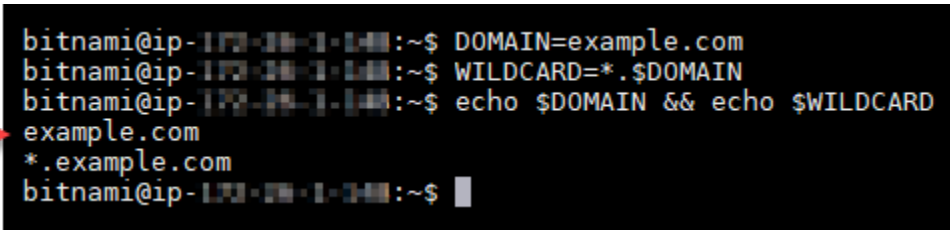
```
DOMAIN=example.com
```

```
WILDCARD=*.$DOMAIN
```

2. Enter the following command to confirm the variables return the correct values:

```
echo $DOMAIN && echo $WILDCARD
```

You should see a result similar to the following:

A terminal window screenshot with a black background and white text. The prompt is 'bitnami@ip-172-31-1-101:~\$'. The user enters 'DOMAIN=example.com', then 'WILDCARD=*. \$DOMAIN', and finally 'echo \$DOMAIN && echo \$WILDCARD'. The output shows 'example.com' and '*.example.com' on separate lines. A red arrow points to the first line of output. The prompt returns to 'bitnami@ip-172-31-1-101:~\$' at the end.

```
bitnami@ip-172-31-1-101:~$ DOMAIN=example.com
bitnami@ip-172-31-1-101:~$ WILDCARD=*. $DOMAIN
bitnami@ip-172-31-1-101:~$ echo $DOMAIN && echo $WILDCARD
example.com
*.example.com
bitnami@ip-172-31-1-101:~$
```

3. Enter the following command to start Certbot in interactive mode. This command tells Certbot to use a manual authorization method with DNS challenges to verify domain ownership. It requests a wildcard certificate for your top-level domain, as well as its subdomains.

```
sudo certbot -d $DOMAIN -d $WILDCARD --manual --preferred-challenges dns certonly
```

4. Enter your email address when prompted, because it's used for renewal and security notices.
5. Read the Let's Encrypt terms of service. When done, press A if you agree. If you disagree, you cannot obtain a Let's Encrypt certificate.
6. Respond accordingly to the prompt to share your email address and to the warning about your IP address being logged.
7. Let's Encrypt now prompts you to verify that you own the domain specified. You do this by adding TXT records to the DNS records for your domain. A set of TXT record values are provided as shown in the following example:

Note

Let's Encrypt may provide a single or multiple TXT records that you must use for verification. In this example, we were provided with two TXT records to use for verification.

```
-----  
Please deploy a DNS TXT record under the name  
_acme-challenge.example.com with the following value:  
9vuaf232Bz0War8BUx3dTNBDpo6lm_4CDX4fpx4reoo  
Before continuing, verify the record is deployed.  
Press Enter to Continue  
-----  
Please deploy a DNS TXT record under the name  
_acme-challenge.example.com with the following value:  
BVkHWl1a0Zhi2UB4BfoSmJV-B_fiSrWfdaf8eBA30dU  
Before continuing, verify the record is deployed.  
-----
```

8. Keep the Lightsail browser-based SSH session open—you return to it later in this tutorial. Continue to the [next section](#) of this tutorial.

Step 4: Add TXT records to your domain's DNS zone

Adding a TXT record to your domain's DNS zone verifies that you own the domain. For demonstration purposes, we use the Lightsail DNS zone. However, the steps might be similar for other DNS zones typically hosted by domain registrars.

Note

To learn more about how to create a Lightsail DNS zone for your domain, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

To add TXT records to your domain's DNS zone in Lightsail

1. On the Lightsail home page, choose the **Domains & DNS** tab.
2. Under the **DNS zones** section of the page, choose the DNS Zone for the domain that you specified in the Certbot certificate request.
3. In the DNS zone editor, choose **DNS records**.
4. Choose **Add record**.

5. In the **Record type** drop-down menu, choose **TXT record**.
6. Enter the values specified by the Let's Encrypt certificate request into the **Record name** and **Responds with** fields.

Note

The Lightsail console pre-populates the apex portion of your domain. For example, if you want to add the `_acme-challenge.example.com` subdomain, then you only have to enter `_acme-challenge` into the text box, and Lightsail adds the `.example.com` portion for you when you save the record.

7. Choose **Save**.
8. Repeat steps 4 through 7 to add the second set of TXT records specified by the Let's Encrypt certificate request.
9. Keep the Lightsail console browser window open—you return to it later in this tutorial. Continue to the [next section](#) of this tutorial.

Step 5: Confirm that the TXT records have propagated

Use the MxToolbox utility to confirm that the TXT records have propagated to the internet's DNS. DNS record propagation might take a while depending on your DNS hosting provider, and the configured time to live (TTL) for your DNS records. It is important that you complete this step, and confirm that your TXT records have propagated, before continuing your Certbot certificate request. Otherwise, your certificate request fails.

To confirm the TXT records have propagated to the internet's DNS

1. Open a new browser window and go to <https://mxtoolbox.com/TXTLookup.aspx>.
2. Enter the following text into the text box.

```
_acme-challenge.Domain
```

Replace *Domain* with your registered domain name.

Example:

```
_acme-challenge.example.com
```

The screenshot shows the MX Toolbox website's 'DNS Text Lookup' page. At the top, there is a navigation bar with a home icon and links for 'MX Lookup', 'Blacklists', 'Diagnostics', and 'Domain Health'. Below this, the main heading is 'DNS Text Lookup'. A search bar contains the domain '_acme-challenge.example.com', and an orange button labeled 'TXT Lookup' is positioned to its right.

3. Choose **TXT Lookup** to run the check.
4. One of the following responses occurs:
 - If your TXT records have propagated to the internet's DNS, you see a response similar to the one shown in the following screenshot. Close the browser window and continue to the [next section](#) of this tutorial.

The screenshot displays the results of a DNS lookup for the domain `txt:_acme-challenge.example.com`. At the top, there is a green 'Find Problems' button and a refresh icon labeled 'txt'. Below this is a table of DNS records:

Type	Domain Name	TTL	Record
TXT	_acme-challenge.example.com	60 sec	9vuaf232Bz0War8BUx3dTNBDpo6lm_4CDX4fpx4reoo
TXT	_acme-challenge.example.com	60 sec	BVkHW11aOZhi2UB4BfoSmJV-B_fiSrwfdaf8eBA30dU

Below the records is a summary table:

	Test	Result
✓	DNS Record Published	DNS Record found

At the bottom, there is a message: "Your DNS hosting provider is 'Amazon Route 53' Need Bulk Dns Provider Data?". Navigation links for 'dns lookup', 'smtp diag', 'blacklist', 'http test', and 'dns propagation' are visible. A footer indicates the report was generated on 10/8/2018 at 8:53:50 PM (UTC 0) and includes a 'Transcript' link.

- If your TXT records have not propagated to the internet's DNS, you see a **DNS Record not found** response. Confirm that you added the correct DNS records to your domains' DNS

zone. If you added the correct records, wait a while longer to let your domain's DNS records propagate, and run the TXT lookup again.

Step 6: Complete the Let's Encrypt SSL certificate request

Go back to the Lightsail browser-based SSH session for your LAMP instance and complete the Let's Encrypt certificate request. Certbot saves your SSL certificate, chain, and key files to a specific directory on your LAMP instance.

To complete the Let's Encrypt SSL certificate request

1. In the Lightsail browser-based SSH session for your LAMP instance, press **Enter** to continue your Let's Encrypt SSL certificate request. If successful, a response similar to the one shown in the following screenshot appears:


```
-----
Please deploy a DNS TXT record under the name
_acme-challenge.example.com with the following value:

9vuaf232Bz0War8BUx3dTNBDpo6lm_4CDX4fpx4reoo

Before continuing, verify the record is deployed.
-----
Press Enter to Continue

-----
Please deploy a DNS TXT record under the name
_acme-challenge.example.com with the following value:

BVkHwll1a0Zhi2UB4BfoSmJV-B_fiSrWfdaf8eBA30dU

Before continuing, verify the record is deployed.
-----
Press Enter to Continue
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/example.com/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/example.com/privkey.pem
  Your cert will expire on 2019-01-06. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-actively renew *all* of your certificates, run
  "certbot renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF:                 https://eff.org/donate-le

bitnami@ip-172-26-1-148:/$ █
```

The message confirms that your certificate, chain, and key files are stored in the `/etc/letsencrypt/live/Domain/` directory. *Domain* will be your registered domain name, such as `/etc/letsencrypt/live/example.com/`.

2. Make note of the expiration date specified in the message. You use it to renew your certificate by that date.

IMPORTANT NOTES:

```

- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/example.com/fullchain.pem
Your key file has been saved at:
  /etc/letsencrypt/live/example.com/privkey.pem
Your cert will expire on 2019-01-06. To obtain a new or tweaked
version of this certificate in the future, simply run certbot
again. To non-interactively renew *all* of your certificates, run
"certbot renew"
- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt:  https://letsencrypt.org/donate
Donating to EFF:                   https://eff.org/donate-le

```

3. Now that you have the Let's Encrypt SSL certificate, continue to the [next section](#) of this tutorial.

Step 7: Create links to the Let's Encrypt certificate files in the Apache server directory

Create links to the Let's Encrypt SSL certificate files in the Apache server directory on your LAMP instance. Also, back up your existing certificates, in case you need them later.

To create links to the Let's Encrypt certificate files in the Apache server directory

1. In the Lightsail browser-based SSH session for your LAMP instance, enter the following command to stop the underlying LAMP stack services:

```
sudo /opt/bitnami/ctlscript.sh stop
```

You should see a response similar to the following:

```

bitnami@ip-100-20-1-101:~$ sudo /opt/bitnami/ctlscript.sh stop
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
/opt/bitnami/php/scripts/ctl.sh : php-fpm stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
bitnami@ip-100-20-1-101:~$

```

2. Enter the following command to set an environment variable for your domain.

```
DOMAIN=Domain
```

In the command, replace *Domain* with your registered domain name.

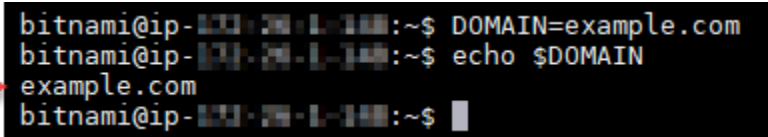
Example:

```
DOMAIN=example.com
```

3. Enter the following command to confirm the variables return the correct values:

```
echo $DOMAIN
```

You should see a result similar to the following:



```
bitnami@ip-172-31-1-144:~$ DOMAIN=example.com  
bitnami@ip-172-31-1-144:~$ echo $DOMAIN  
example.com  
bitnami@ip-172-31-1-144:~$
```

A red arrow points to the output of the `echo $DOMAIN` command, which is `example.com`.

4. Enter the following commands individually to rename your existing certificate files as backups. Refer to the **Important** block at the beginning of this tutorial for information about the different distributions and file structures.

- For Debian Linux distributions

Approach A (Bitnami installations using system packages):

```
sudo mv /opt/bitnami/apache2/conf/bitnami/certs/server.crt /opt/bitnami/apache2/  
conf/bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/apache2/conf/bitnami/certs/server.key /opt/bitnami/apache2/  
conf/bitnami/certs/server.key.old
```

Approach B (Self-contained Bitnami installations):

```
sudo mv /opt/bitnami/apache2/conf/server.crt /opt/bitnami/apache2/conf/  
server.crt.old
```

```
sudo mv /opt/bitnami/apache2/conf/server.key /opt/bitnami/apache2/conf/  
server.key.old
```

- For older instances that use the Ubuntu Linux distribution:

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.crt /opt/bitnami/apache/conf/bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.key /opt/bitnami/apache/conf/bitnami/certs/server.key.old
```

5. Enter the following commands individually to create links to your Let's Encrypt certificate files in the apache2 server directory. Refer to the **Important** block at the beginning of this tutorial for information about the different distributions and file structures.

- For Debian Linux distributions

Approach A (Bitnami installations using system packages):

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache2/conf/bitnami/certs/server.key
```

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache2/conf/bitnami/certs/server.crt
```

Approach B (Self-contained Bitnami installations):

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache2/conf/server.key
```

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache2/conf/server.crt
```

- For older instances that use the Ubuntu Linux distribution:

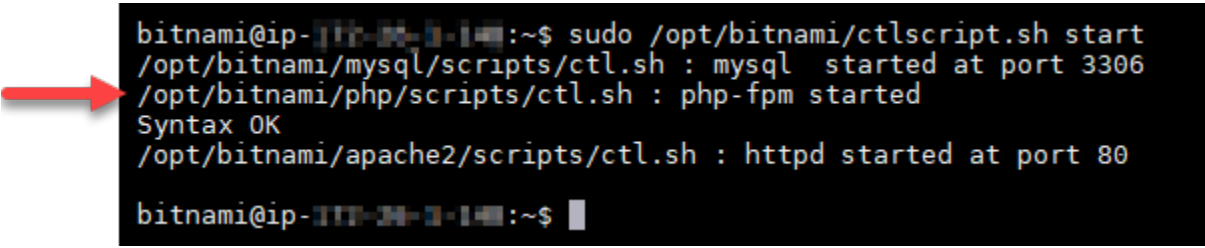
```
sudo ln -s /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache/conf/bitnami/certs/server.key
```

```
sudo ln -s /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache/conf/bitnami/certs/server.crt
```

6. Enter the following command to start the underlying LAMP stack services that you had stopped earlier:

```
sudo /opt/bitnami/ctlscript.sh start
```

You should see a result similar to the following:



```
bitnami@ip-10-10-10-10:~$ sudo /opt/bitnami/ctlscript.sh start
/opt/bitnami/mysql/scripts/ctl.sh : mysql started at port 3306
/opt/bitnami/php/scripts/ctl.sh : php-fpm started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80
bitnami@ip-10-10-10-10:~$
```

Your LAMP instance is now configured to use SSL encryption. However, traffic is not automatically redirected from HTTP to HTTPS.

7. Continue to the [next section](#) of this tutorial.

Step 8: Configure HTTP to HTTPS redirection for your web application

You can configure an HTTP to HTTPS redirect for your LAMP instance. Automatically redirecting from HTTP to HTTPS makes your site accessible only by your customers using SSL, even when they connect using HTTP.

To configure HTTP to HTTPS redirection for your web application

1. In the Lightsail browser-based SSH session for your LAMP instance, enter the following command to edit the Apache web server configuration file using the Vim text editor:

```
sudo vim /opt/bitnami/apache2/conf/bitnami/bitnami.conf
```

Note

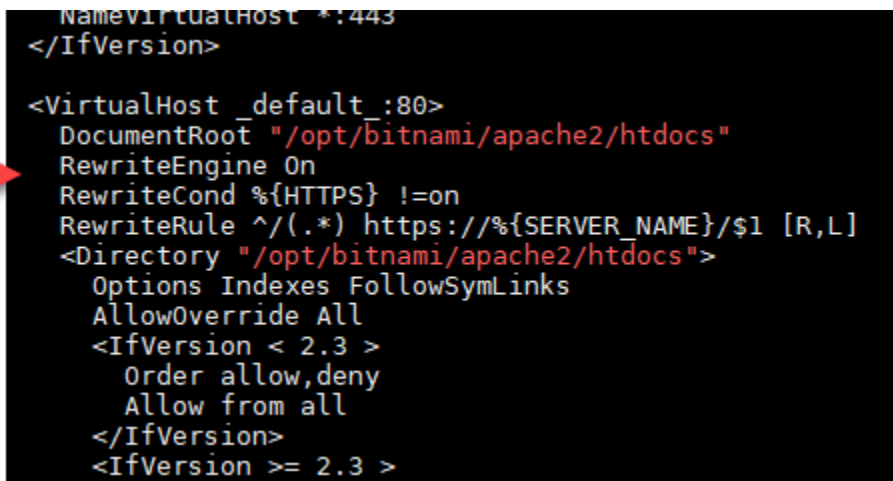
This tutorial uses Vim for demonstration purposes; however, you can use any text editor of your choice for this step.

2. Press `i` to enter insert mode in the Vim editor.
3. In the file, enter the following text between `DocumentRoot "/opt/bitnami/apache2/htdocs"` and `<Directory "/opt/bitnami/apache2/htdocs">`:

```
RewriteEngine On
```

```
RewriteCond %{HTTPS} !=on  
RewriteRule ^/(.*) https://%{SERVER_NAME}/$1 [R,L]
```

The result should look like the following:



```
NameVirtualHost *:443  
</IfVersion>  
  
<VirtualHost _default_:80>  
DocumentRoot "/opt/bitnami/apache2/htdocs"  
RewriteEngine On  
RewriteCond %{HTTPS} !=on  
RewriteRule ^/(.*) https://%{SERVER_NAME}/$1 [R,L]  
<Directory "/opt/bitnami/apache2/htdocs">  
Options Indexes FollowSymLinks  
AllowOverride All  
<IfVersion < 2.3 >  
Order allow,deny  
Allow from all  
</IfVersion>  
<IfVersion >= 2.3 >
```

4. Press the **ESC** key, and then enter `:wq` to write (save) your edits, and quit Vim.
5. Enter the following command to restart the underlying LAMP stack services and make your edits effective:

```
sudo /opt/bitnami/ctlscript.sh restart
```

Your LAMP instance is now configured to automatically redirect connections from HTTP to HTTPS. When a visitor goes to `http://www.example.com`, they are automatically redirected to the encrypted `https://www.example.com` address.

Step 9: Renew the Let's Encrypt certificates every 90 days

Let's Encrypt certificates are valid for 90 days. Certificates can be renewed 30 days before they expire. To renew the Let's Encrypt certificates, run the original command used to obtain them. Repeat the steps in the [Request a Let's Encrypt SSL wildcard certificate](#) section of this tutorial.

Tutorial: Use Let's Encrypt SSL certificates with your Lightsail Nginx instance

Amazon Lightsail makes it easy to secure your websites and applications with SSL/TLS using Lightsail load balancers. However, using a Lightsail load balancer might not generally be the right

choice. Perhaps your site doesn't need the scalability or fault tolerance load balancers provide, or maybe you're optimizing for cost.

In the latter case, you might consider using Let's Encrypt to obtain a free SSL certificate. If so, that's no problem. You can integrate those certificates with Lightsail instances. This tutorial shows you how to request a Let's Encrypt wildcard certificate using Certbot, and integrate it with your Nginx instance.

Important

- The Linux distribution used by Bitnami instances changed from Ubuntu to Debian in July, 2020. Because of this change, some of the steps in this tutorial will differ depending on the Linux distribution of your instance. All Bitnami blueprint instances created after the change use the Debian Linux distribution. Instances created before the change will continue to use the Ubuntu Linux distribution. To check the distribution of your instance, run the `uname -a` command. The response will show either Ubuntu or Debian as your instance's Linux distribution.
- Bitnami is in the process of modifying the file structure for many of their stacks. The file paths in this tutorial may change depending on whether your Bitnami stack uses native Linux system packages (Approach A), or if it is a self-contained installation (Approach B). To identify your Bitnami installation type and which approach to follow, run the following command:

```
test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach A: Using system packages." || echo "Approach B: Self-contained installation."
```

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Install Certbot on your Lightsail instance](#)
- [Step 3: Request a Let's Encrypt SSL wildcard certificate](#)
- [Step 4: Add TXT records to your domain's DNS zone](#)
- [Step 5: Confirm that the TXT records have propagated](#)
- [Step 6: Complete the Let's Encrypt SSL certificate request](#)

- [Step 7: Create links to the Let's Encrypt certificate files in the Nginx server directory](#)
- [Step 8: Configure HTTP to HTTPS redirection for your web application](#)
- [Step 9: Renew the Let's Encrypt certificates every 90 days](#)

Step 1: Complete the prerequisites

Complete the following prerequisites if you haven't already done so:

- Create a Nginx instance in Lightsail. To learn more, see [Create an instance](#).
- Register a domain name, and get administrative access to edit its DNS records. To learn more, see [DNS](#).

Note

We recommend that you manage your domain's DNS records using a Lightsail DNS zone. To learn more, see [Create a DNS zone to manage your domain's DNS records](#).

- Use the browser-based SSH terminal in the Lightsail console to perform the steps in this tutorial. However, you can also use your own SSH client, such as PuTTY. To learn more about configuring PuTTY, see [Download and set up PuTTY to connect using SSH in Amazon Lightsail](#).

After you've completed the prerequisites, continue to the [next section](#) of this tutorial.


Step 2: Install Certbot on your Lightsail instance

Certbot is a client used to request a certificate from Let's Encrypt and deploy it to a web server. Let's Encrypt uses the ACME protocol to issue certificates, and Certbot is an ACME-enabled client that interacts with Let's Encrypt.

To install Certbot on your Lightsail instance

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the SSH quick connect icon for the instance that you want to connect to.

5. Enter the following command to add Certbot to the local apt repository:

 **Note**

Step 5 applies only to instances that use the Ubuntu Linux distribution. Skip this step if your instance uses the Debian Linux distribution.

```
sudo apt-add-repository ppa:certbot/certbot -y
```

6. Enter the following command to update apt to include the new repository:

```
sudo apt-get update -y
```

7. Enter the following command to install Certbot:

```
sudo apt-get install certbot -y
```

Certbot is now installed on your Lightsail instance.

8. Keep the browser-based SSH terminal window open—you return to it later in this tutorial. Continue to the [next section](#) of this tutorial.

Step 3: Request a Let's Encrypt SSL wildcard certificate

Begin the process of requesting a certificate from Let's Encrypt. Using Certbot, request a wildcard certificate, which lets you use a single certificate for a domain and its subdomains. For example, a single wildcard certificate works for the `example.com` top-level domain, and the `blog.example.com`, and `stuff.example.com` subdomains.

To request a Let's Encrypt SSL wildcard certificate

1. In the same browser-based SSH terminal window used in [step 2](#) of this tutorial, enter the following commands to set an environment variable for your domain. You can now more efficiently copy and paste commands to obtain the certificate. Be sure to replace *domain* with the name of your registered domain name.

```
DOMAIN=domain
```

```
WILDCARD=*. $DOMAIN
```

Example:

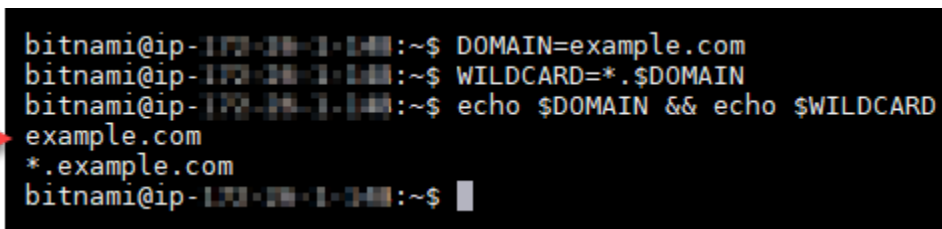
```
DOMAIN=example.com
```

```
WILDCARD=*. $DOMAIN
```

2. Enter the following command to confirm the variables return the correct values:

```
echo $DOMAIN && echo $WILDCARD
```

You should see a result similar to the following:



```
bitnami@ip-172-31-1-101:~$ DOMAIN=example.com
bitnami@ip-172-31-1-101:~$ WILDCARD=*. $DOMAIN
bitnami@ip-172-31-1-101:~$ echo $DOMAIN && echo $WILDCARD
example.com
*.example.com
bitnami@ip-172-31-1-101:~$
```

A red arrow points to the output of the echo command, which is "example.com".

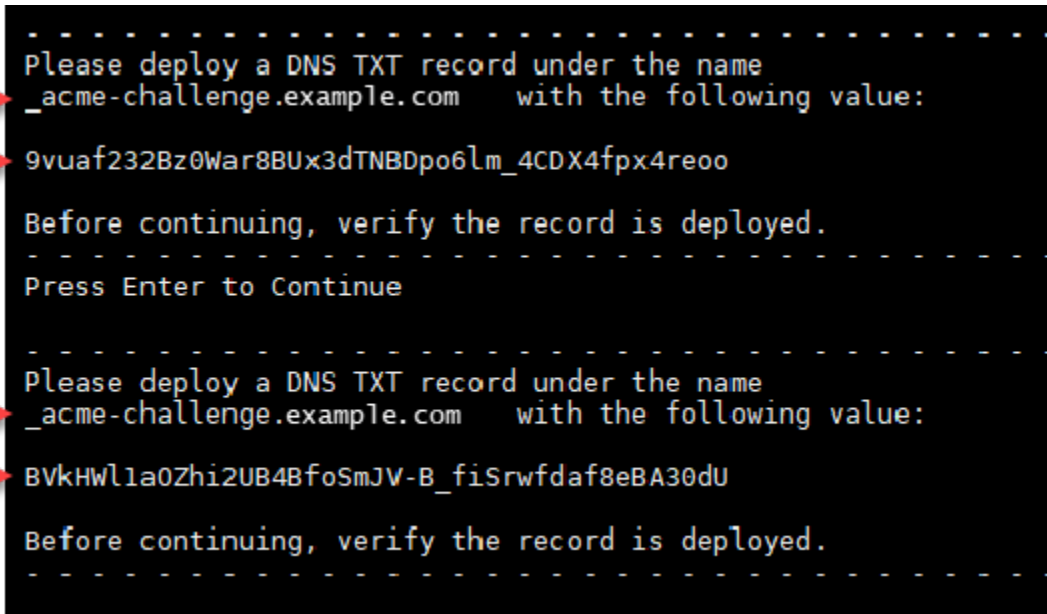
3. Enter the following command to start Certbot in interactive mode. This command tells Certbot to use a manual authorization method with DNS challenges to verify domain ownership. It requests a wildcard certificate for your top-level domain, as well as its subdomains.

```
sudo certbot -d $DOMAIN -d $WILDCARD --manual --preferred-challenges dns certonly
```

4. Enter your email address when prompted, because it's used for renewal and security notices.
5. Read the Let's Encrypt terms of service. When done, press A if you agree. If you disagree, you cannot obtain a Let's Encrypt certificate.
6. Respond accordingly to the prompt to share your email address and to the warning about your IP address being logged.
7. Let's Encrypt now prompts you to verify that you own the domain specified. You do this by adding TXT records to the DNS records for your domain. A set of TXT record values are provided as shown in the following example:

Note

Let's Encrypt may provide a single or multiple TXT records that you must use for verification. In this example, we were provided with two TXT records to use for verification.



```
-----  
Please deploy a DNS TXT record under the name  
_acme-challenge.example.com with the following value:  
9vuaf232Bz0War8BUx3dTNBDpo6lm_4CDX4fpx4reoo  
Before continuing, verify the record is deployed.  
Press Enter to Continue  
-----  
Please deploy a DNS TXT record under the name  
_acme-challenge.example.com with the following value:  
BVkHWl1a0Zhi2UB4BfoSmJV-B_fiSrwfdaF8eBA30dU  
Before continuing, verify the record is deployed.  
-----
```

8. Keep the Lightsail browser-based SSH session open—you return to it later in this tutorial. Continue to the [next section](#) of this tutorial.

Step 4: Add TXT records to your domain's DNS zone

Adding a TXT record to your domain's DNS zone verifies that you own the domain. For demonstration purposes, we use the Lightsail DNS zone. However, the steps might be similar for other DNS zones typically hosted by domain registrars.

Note

To learn more about how to create a Lightsail DNS zone for your domain, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

To add TXT records to your domain's DNS zone in Lightsail

1. On the Lightsail home page, choose the **Domains & DNS** tab.
2. Under the **DNS zones** section of the page, choose the DNS Zone for the domain that you specified in the Certbot certificate request.
3. In the DNS zone editor, choose **DNS records**.
4. Choose **Add record**.
5. In the **Record type** drop-down menu, choose **TXT record**.
6. Enter the values specified by the Let's Encrypt certificate request into the **Record name** and **Responds with** fields.

Note

The Lightsail console pre-populates the apex portion of your domain. For example, if you want to add the `_acme-challenge.example.com` subdomain, then you only have to enter `_acme-challenge` into the text box, and Lightsail adds the `.example.com` portion for you when you save the record.

7. Choose **Save**.
8. Repeat steps 4 through 7 to add the second set of TXT records specified by the Let's Encrypt certificate request.
9. Keep the Lightsail console browser window open—you return to it later in this tutorial. Continue to the [next section](#) of this tutorial.

Step 5: Confirm that the TXT records have propagated

Use the MxToolbox utility to confirm that the TXT records have propagated to the Internet's DNS. DNS record propagation might take a while depending on your DNS hosting provider, and the configured time to live (TTL) for your DNS records. It is important that you complete this step, and confirm that your TXT records have propagated, before continuing your Certbot certificate request. Otherwise, your certificate request fails.

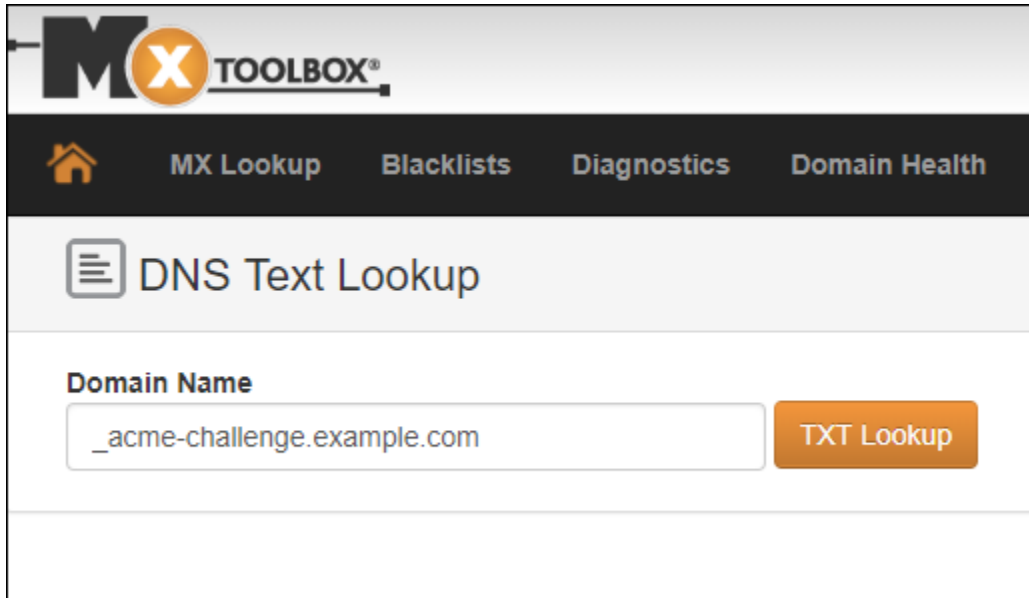
To confirm the TXT records have propagated to the Internet's DNS

1. Open a new browser window and go to <https://mxtoolbox.com/TXTLookup.aspx>.
2. Enter the following text into the text box. Be sure to replace `domain` with your domain.

`_acme-challenge.domain`

Example:

`_acme-challenge.example.com`



3. Choose **TXT Lookup** to run the check.
4. One of the following responses occurs:
 - If your TXT records have propagated to the Internet's DNS, you see a response similar to the one shown in the following screenshot. Close the browser window and continue to the [next section](#) of this tutorial.

txt:_acme-challenge.example.com [Find Problems](#) [txt](#)

Type	Domain Name	TTL	Record
TXT	_acme-challenge.example.com	60 sec	9vuaf232Bz0War8BUx3dTNBDpo6lm_4CDX4fpx4reoo
TXT	_acme-challenge.example.com	60 sec	BVkHW11a0Zhi2UB4BfoSmJV-B_fiSrwfdaf8eBA30dU

	Test	Result
✓	DNS Record Published	DNS Record found

Your DNS hosting provider is "Amazon Route 53" [Need Bulk Dns Provider Data?](#)

[dns lookup](#)
[smtp diag](#)
[blacklist](#)
[http test](#)
[dns propagation](#)

Reported by on 10/8/2018 at 8:53:50 PM (UTC 0), [just for you.](#) [Transcript](#)

- If your TXT records have not propagated to the Internet's DNS, you see a **DNS Record not found** response. Confirm that you added the correct DNS records to your domains' DNS zone. If you added the correct records, wait a while longer to let your domain's DNS records propagate, and run the TXT lookup again.

Step 6: Complete the Let's Encrypt SSL certificate request

Go back to the Lightsail browser-based SSH session for your Nginx instance and complete the Let's Encrypt certificate request. Certbot saves your SSL certificate, chain, and key files to a specific directory on your Nginx instance.

To complete the Let's Encrypt SSL certificate request

1. In the Lightsail browser-based SSH session for your Nginx instance, press **Enter** to continue your Let's Encrypt SSL certificate request. If successful, a response similar to the one shown in the following screenshot appears:

```
-----
Please deploy a DNS TXT record under the name
_acme-challenge.example.com with the following value:

9vuaf232Bz0War8BUx3dTNBDpo6lm_4CDX4fpx4reoo

Before continuing, verify the record is deployed.
-----
Press Enter to Continue

-----
Please deploy a DNS TXT record under the name
_acme-challenge.example.com with the following value:

BVkHwll1a0Zhi2UB4BfoSmJV-B_fiSrWfdaf8eBA30dU

Before continuing, verify the record is deployed.
-----
Press Enter to Continue
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/example.com/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/example.com/privkey.pem
  Your cert will expire on 2019-01-06. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-actively renew *all* of your certificates, run
  "certbot renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF: https://eff.org/donate-le

bitnami@ip-172-26-1-148:/$
```

The message confirms that your certificate, chain, and key files are stored in the `/etc/letsencrypt/live/domain/` directory. Make sure to replace *domain* with your domain, such as `/etc/letsencrypt/live/example.com/`.

2. Make note of the expiration date specified in the message. You use it to renew your certificate by that date.


```

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/example.com/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/example.com/privkey.pem
  Your cert will expire on 2019-01-06. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-interactively renew *all* of your certificates, run
  "certbot renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt:  https://letsencrypt.org/donate
  Donating to EFF:                   https://eff.org/donate-le

```

- Now that you have the Let's Encrypt SSL certificate, continue to the [next section](#) of this tutorial.

Step 7: Create links to the Let's Encrypt certificate files in the Nginx server directory

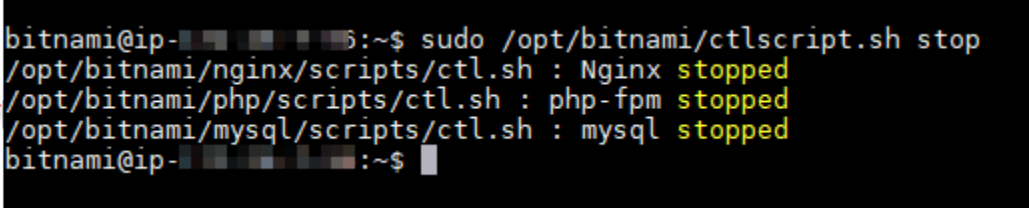
Create links to the Let's Encrypt SSL certificate files in the Nginx server directory on your Nginx instance. Also, back up your existing certificates, in case you need them later.

To create links to the Let's Encrypt certificate files in the Nginx server directory

- In the Lightsail browser-based SSH session for your Nginx instance, enter the following command to stop the underlying services:

```
sudo /opt/bitnami/ctlscript.sh stop
```

You should see a response similar to the following:



```

bitnami@ip-...:~$ sudo /opt/bitnami/ctlscript.sh stop
/opt/bitnami/nginx/scripts/ctl.sh : Nginx stopped
/opt/bitnami/php/scripts/ctl.sh : php-fpm stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
bitnami@ip-...:~$

```

- Enter the following command to set an environment variable for your domain. You can more efficiently copy and paste commands to link the certificate files. Be sure to replace *domain* with the name of your registered domain.

```
DOMAIN=domain
```

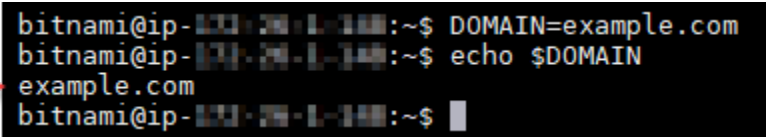
Example:

```
DOMAIN=example.com
```

3. Enter the following command to confirm the variables return the correct values:

```
echo $DOMAIN
```

You should see a result similar to the following:



```
bitnami@ip-172-31-1-100:~$ DOMAIN=example.com  
bitnami@ip-172-31-1-100:~$ echo $DOMAIN  
example.com  
bitnami@ip-172-31-1-100:~$
```

A red arrow points to the output of the second command, `example.com`.

4. Enter the following commands individually to rename your existing certificate files as backups. Refer to the **Important** block at the beginning of this tutorial for information about the different distributions and file structures.

- For Debian Linux distributions

Approach A (Bitnami installations using system packages):

```
sudo mv /opt/bitnami/nginx/conf/bitnami/certs/server.crt /opt/bitnami/nginx/conf/  
bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/nginx/conf/bitnami/certs/server.key /opt/bitnami/nginx/conf/  
bitnami/certs/server.key.old
```

Approach B (Self-contained Bitnami installations):

```
sudo mv /opt/bitnami/nginx/conf/server.crt /opt/bitnami/nginx/conf/server.crt.old
```

```
sudo mv /opt/bitnami/nginx/conf/server.key /opt/bitnami/nginx/conf/server.key.old
```

- For older instances that use the Ubuntu Linux distribution:

```
sudo mv /opt/bitnami/nginx/conf/bitnami/certs/server.crt /opt/bitnami/nginx/conf/  
bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/nginx/conf/bitnami/certs/server.key /opt/bitnami/nginx/conf/bitnami/certs/server.key.old
```

5. Enter the following commands individually to create links to your Let's Encrypt certificate files in the Nginx server directory. Refer to the **Important** block at the beginning of this tutorial for information about the different distributions and file structures.

- For Debian Linux distributions

Approach A (Bitnami installations using system packages):

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/nginx/conf/bitnami/certs/server.key
```

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/nginx/conf/bitnami/certs/server.crt
```

Approach B (Self-contained Bitnami installations):

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/nginx/conf/server.key
```

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/nginx/conf/server.crt
```

- For older instances that use the Ubuntu Linux distribution:

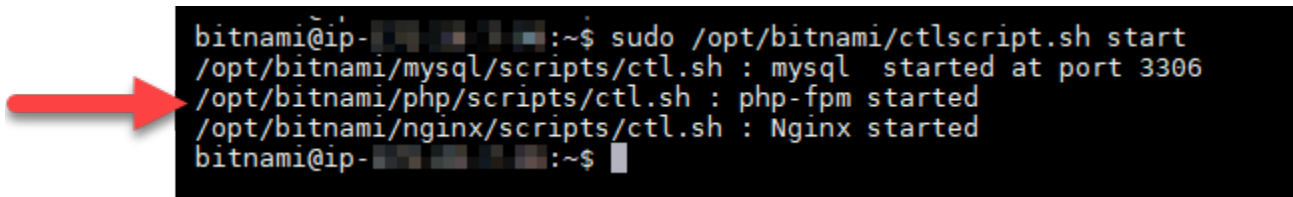
```
sudo ln -s /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/nginx/conf/bitnami/certs/server.key
```

```
sudo ln -s /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/nginx/conf/bitnami/certs/server.crt
```

6. Enter the following command to start the underlying services that you stopped earlier:

```
sudo /opt/bitnami/ctlscript.sh start
```

You should see a result similar to the following:



```
bitnami@ip-...:~$ sudo /opt/bitnami/ctlscript.sh start
/opt/bitnami/mysql/scripts/ctl.sh : mysql started at port 3306
/opt/bitnami/php/scripts/ctl.sh : php-fpm started
/opt/bitnami/nginx/scripts/ctl.sh : Nginx started
bitnami@ip-...:~$
```

Your Nginx instance is now configured to use SSL encryption. However, traffic is not automatically redirected from HTTP to HTTPS.

7. Continue to the [next section](#) of this tutorial.

Step 8: Configure HTTP to HTTPS redirection for your web application

You can configure an HTTP to HTTPS redirect for your Nginx instance. Automatically redirecting from HTTP to HTTPS makes your site accessible only by your customers using SSL, even when they connect using HTTP. Refer to the Important block at the beginning of this tutorial for information about the different distributions and file structures.

This tutorial uses Vim for demonstration purposes; however, you can use any text editor of your choice.

For Debian Linux distributions – Configure HTTP to HTTPS redirection for your web application

1. In the Lightsail browser-based SSH session for your Nginx instance, enter the following command to modify the server-block configuration file. Replace `<ApplicationName>` with the name of your application.

```
sudo vim /opt/bitnami/nginx/conf/server_blocks/<ApplicationName>-server-block.conf
```

2. Press `i` to enter insert mode in the Vim editor.
3. Edit the file with the information from the following example:

```
server {
    listen 80 default_server;
    root /opt/bitnami/APPNAME;
    return 301 https://$host$request_uri;
}
```

4. Press the **ESC** key, and then enter `:wq` to write (save) your edits, and quit Vim.
5. Enter the following command to modify the server section of the Nginx configuration file:

```
sudo vim /opt/bitnami/nginx/conf/nginx.conf
```

6. Press **i** to enter insert mode in the Vim editor.
7. Edit the file with the information from the following example:

```
server {  
    listen 80;  
    server_name localhost;  
    return 301 https://$host$request_uri;  
}
```

8. Press the **ESC** key, and then enter `:wq` to write (save) your edits, and quit Vim.
9. Enter the following command to restart the underlying services and make your edits effective:

```
sudo /opt/bitnami/ctlscript.sh restart
```

Approach B (Self-contained Bitnami installations):

1. In the Lightsail browser-based SSH session for your Nginx instance, enter the following command to modify the server section of the Nginx configuration file:

```
sudo vim /opt/bitnami/nginx/conf/nginx.conf
```

2. Press **i** to enter insert mode in the Vim editor.
3. Edit the file with the information from the following example:

```
server {  
    listen 80;  
    server_name localhost;  
    return 301 https://$host$request_uri;  
}
```

4. Press the **ESC** key, and then enter `:wq` to write (save) your edits, and quit Vim.
5. Enter the following command to restart the underlying services and make your edits effective:

```
sudo /opt/bitnami/ctlscript.sh restart
```

For older instances that use the Ubuntu Linux distribution – Configure HTTP to HTTPS redirection for your web application

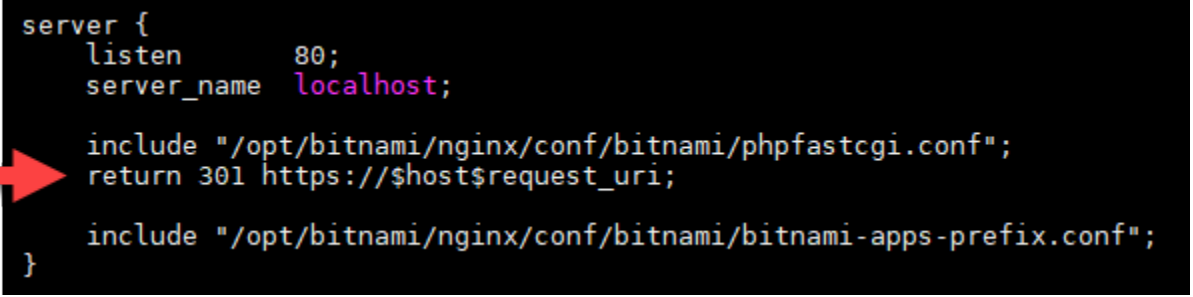
1. In the Lightsail browser-based SSH session for your Nginx instance, enter the following command to edit the Nginx web server configuration file using the Vim text editor:

```
sudo vim /opt/bitnami/nginx/conf/bitnami/bitnami.conf
```

2. Press `i` to enter insert mode in the Vim editor.
3. In the file, enter the following text between `server_name localhost;` and `include "/opt/bitnami/nginx/conf/bitnami/bitnami-apps-prefix.conf";`:

```
return 301 https://$host$request_uri;
```

The result should look like the following:



```
server {
    listen      80;
    server_name localhost;

    include "/opt/bitnami/nginx/conf/bitnami/phpfastcgi.conf";
    return 301 https://$host$request_uri;

    include "/opt/bitnami/nginx/conf/bitnami/bitnami-apps-prefix.conf";
}
```

A red arrow points to the line `return 301 https://$host$request_uri;` in the screenshot.

4. Press the **ESC** key, and then enter `:wq` to write (save) your edits, and quit Vim.
5. Enter the following command to restart the underlying services and make your edits effective:

```
sudo /opt/bitnami/ctlscript.sh restart
```

Your Nginx instance is now configured to automatically redirect connections from HTTP to HTTPS. When a visitor goes to `http://www.example.com`, they are automatically redirected to the encrypted `https://www.example.com` address.

Step 9: Renew the Let's Encrypt certificates every 90 days

Let's Encrypt certificates are valid for 90 days. Certificates can be renewed 30 days before they expire. To renew the Let's Encrypt certificates, run the original command used to obtain them. Repeat the steps in the [Request a Let's Encrypt SSL wildcard certificate](#) section of this tutorial.

Tutorial: Use Let's Encrypt SSL certificates with your Lightsail WordPress instance

Tip

Lightsail offers a guided workflow that automates the installation and configuration of a Let's Encrypt certificate on your WordPress instance. We highly recommend that you use the workflow instead of following the manual steps in this tutorial. For more information, see [Launch and configure a WordPress instance](#).

Amazon Lightsail makes it easy to secure your websites and applications with SSL/TLS using Lightsail load balancers. However, using a Lightsail load balancer might not generally be the right choice. Perhaps your site doesn't need the scalability or fault tolerance that load balancers provide, or maybe you're optimizing for cost. In the latter case, you might consider using Let's Encrypt to obtain a free SSL certificate. If so, that's no problem. You can integrate those certificates with Lightsail instances.

With this guide, you'll learn how to request a Let's Encrypt wildcard certificate using Certbot, and integrate it with your WordPress instance using the Really Simple SSL plugin.

- The Linux distribution used by Bitnami instances changed from Ubuntu to Debian in July, 2020. Because of this change, some of the steps in this tutorial will differ depending on the Linux distribution of your instance. All Bitnami blueprint instances created after the change use the Debian Linux distribution. Instances created before the change will continue to use the Ubuntu Linux distribution. To check the distribution of your instance, run the `uname -a` command. The response will show either Ubuntu or Debian as your instance's Linux distribution.
- Bitnami has modified the file structure for many of their stacks. The file paths in this tutorial may change depending on whether your Bitnami stack uses native Linux system packages (Approach A), or if it is a self-contained installation (Approach B). To identify your Bitnami installation type and which approach to follow, run the following command:

```
test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach A: Using system packages." || echo "Approach B: Self-contained installation."
```

Contents

- [Before getting started](#)
- [Step 1: Complete the prerequisites](#)
- [Step 2: Install Certbot on your Lightsail instance](#)
- [Step 3: Request a Let's Encrypt SSL wildcard certificate](#)
- [Step 4: Add TXT records to your domain's DNS zone](#)
- [Step 5: Confirm that the TXT records have propagated](#)
- [Step 6: Complete the Let's Encrypt SSL certificate request](#)
- [Step 7: Create links to the Let's Encrypt certificate files in the Apache server directory](#)
- [Step 8: Integrate the SSL certificate with your WordPress site using the Really Simple SSL plug-in](#)
- [Step 9: Renew the Let's Encrypt certificates every 90 days](#)

Before getting started

You should consider the following before getting started with this tutorial:

Use the Bitnami HTTPS configuration (bncert) tool instead

The steps outlined in this tutorial show you how to implement an SSL/TLS certificate using a manual process. However, Bitnami offers a more automated process that uses the Bitnami HTTPS configuration (bncert) tool that is typically pre-installed on WordPress instances in Lightsail. We highly recommend that you use that tool instead of following the manual steps in this tutorial. This tutorial was written before the bncert tool became available. For more information about using the bncert tool, see [Enabling HTTPS on your WordPress instance in Amazon Lightsail](#).

Identify the Linux distribution of your WordPress instance

The Linux distribution used by Bitnami instances changed from Ubuntu to Debian in July, 2020. All Bitnami blueprint instances created after the change use the Debian Linux distribution. Instances created before the change will continue to use the Ubuntu Linux distribution. Because of this change, some of the steps in this tutorial will differ depending on the Linux distribution of your instance. You must identify the Linux distribution of your instance so that you know which steps in this tutorial to use. To identify the Linux distribution of your instance, run the `uname -a` command. The response will show either Ubuntu or Debian as your instance's Linux distribution.

Identify the tutorial approach that applies to your instance

Bitnami is in the process of modifying the file structure for many of their stacks. The file paths in this tutorial may change depending on whether your Bitnami stack uses native Linux system packages (Approach A), or if it is a self-contained installation (Approach B). To identify your Bitnami installation type and which approach to follow, run the following command:

```
test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach A: Using system packages." || echo "Approach B: Self-contained installation."
```

Step 1: Complete the prerequisites

Complete the following prerequisites if you haven't already done so:

- Create a WordPress instance in Lightsail. To learn more, see [Create an instance](#).
- Register a domain name, and get administrative access to edit its DNS records. To learn more, see [DNS](#).

We recommend that you manage your domain's DNS records using a Lightsail DNS zone. To learn more, see [Create a DNS zone to manage your domain's DNS records](#).

- Use the browser-based SSH terminal in the Lightsail console to perform the steps in this tutorial. However, you can also use your own SSH client, such as PuTTY. To learn more about configuring PuTTY, see [Download and set up PuTTY to connect using SSH in Amazon Lightsail](#).

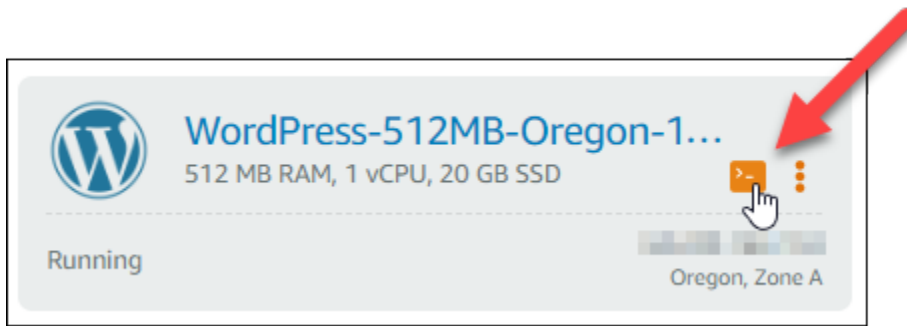
After you've completed the prerequisites, continue to the [next section](#) of this tutorial.

Step 2: Install Certbot on your Lightsail instance

Certbot is a client used to request a certificate from Let's Encrypt and deploy it to a web server. Let's Encrypt uses the ACME protocol to issue certificates, and Certbot is an ACME-enabled client that interacts with Let's Encrypt.

To install Certbot on your Lightsail instance

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the SSH quick connect icon for the instance that you want to connect to.



3. After your Lightsail browser-based SSH session is connected, enter the following command to update the packages on your instance:

```
sudo apt-get update
```

```
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1070-aws x86_64)

      _ _ _
     / _ \
    / ___ \
   / ___ \
  / ___ \
 / ___ \
/_/___ \

*** Welcome to the Bitnami WordPress 4.9.8-0 ***
*** Documentation: https://docs.bitnami.com/aws/apps/wordpress/ ***
***                 https://docs.bitnami.com/aws/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***
Last login: Tue Oct 30 14:22:03 2018 from [redacted]
bitnami@ip-[redacted]:~$ sudo apt-get update
```


4. Enter the following command to install the software properties package. Certbot's developers use a Personal Package Archive (PPA) to distribute Certbot. The software properties package makes it more efficient to work with PPAs.

```
sudo apt-get install software-properties-common
```

Note

If you encounter a `Could not get lock` error when running the `sudo apt-get install` command, please wait approximately 15 minutes and try again. This error may be caused by a cron job that is using the Apt package management tool to install unattended upgrades.

5. Enter the following commands to install the GPG package, and add Certbot to the local apt repository:

 **Note**

Step 5 applies only to instances that use the Ubuntu Linux distribution. Skip this step if your instance uses the Debian Linux distribution.

```
sudo apt-get install gpg -y
```

```
sudo apt-add-repository ppa:certbot/certbot -y
```

6. Enter the following command to update apt to include the new repository:

```
sudo apt-get update -y
```

7. Enter the following command to install Certbot:

```
sudo apt-get install certbot -y
```

Certbot is now installed on your Lightsail instance.

8. Keep the browser-based SSH terminal window open—you return to it later in this tutorial. Continue to the [next section](#) of this tutorial.

Step 3: Request a Let's Encrypt SSL wildcard certificate

Begin the process of requesting a certificate from Let's Encrypt. Using Certbot, request a wildcard certificate, which lets you use a single certificate for a domain and its subdomains. For example, a single wildcard certificate works for the `example.com` top-level domain, and the `blog.example.com`, and `stuff.example.com` subdomains.

To request a Let's Encrypt SSL wildcard certificate

1. In the same browser-based SSH terminal window used in [step 2](#) of this tutorial, enter the following commands to set an environment variable for your domain. You can now more

efficiently copy and paste commands to obtain the certificate. Be sure to replace *domain* with the name of your registered domain.

```
DOMAIN=domain
```

```
WILDCARD=*.$DOMAIN
```

Example:

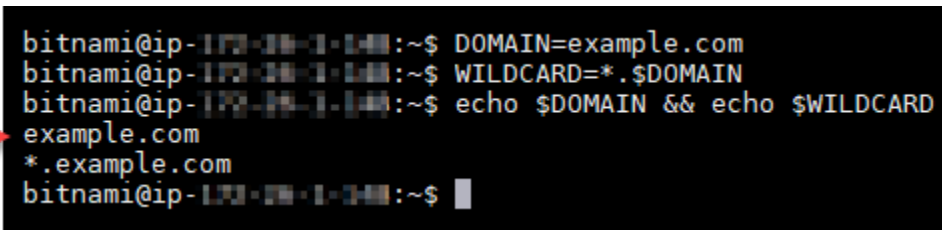
```
DOMAIN=example.com
```

```
WILDCARD=*.$DOMAIN
```

2. Enter the following command to confirm the variables return the correct values:

```
echo $DOMAIN && echo $WILDCARD
```

You should see a result similar to the following:



```
bitnami@ip-172-31-1-141:~$ DOMAIN=example.com
bitnami@ip-172-31-1-141:~$ WILDCARD=*.$DOMAIN
bitnami@ip-172-31-1-141:~$ echo $DOMAIN && echo $WILDCARD
example.com
*.example.com
bitnami@ip-172-31-1-141:~$ █
```

A red arrow points to the output of the command, which is *example.com*.

3. Enter the following command to start Certbot in interactive mode. This command tells Certbot to use a manual authorization method with DNS challenges to verify domain ownership. It requests a wildcard certificate for your top-level domain, as well as its subdomains.

```
sudo certbot -d $DOMAIN -d $WILDCARD --manual --preferred-challenges dns certonly
```

4. Enter your email address when prompted, because it's used for renewal and security notices.
5. Read the Let's Encrypt terms of service. When done, press A if you agree. If you disagree, you cannot obtain a Let's Encrypt certificate.
6. Respond accordingly to the prompt to share your email address and to the warning about your IP address being logged.

- Let's Encrypt now prompts you to verify that you own the domain specified. You do this by adding TXT records to the DNS records for your domain. A set of TXT record values are provided as shown in the following example:

Note

Let's Encrypt may provide a single or multiple TXT records that you must use for verification. In this example, we were provided with two TXT records to use for verification.

```
-----  
Please deploy a DNS TXT record under the name  
_acme-challenge.example.com with the following value:  
9vuaf232Bz0War8BUx3dTnBDpo6lm_4CDX4fpx4reoo  
Before continuing, verify the record is deployed.  
Press Enter to Continue  
-----  
Please deploy a DNS TXT record under the name  
_acme-challenge.example.com with the following value:  
BVkHWl1a0Zhi2UB4BfoSmJV-B_fiSrWfdaf8eBA30dU  
Before continuing, verify the record is deployed.  
-----
```

- Keep the Lightsail browser-based SSH session open—you return to it later in this tutorial. Continue to the [next section](#) of this tutorial.

Step 4: Add TXT records to your domain's DNS zone

Adding a TXT record to your domain's DNS zone verifies that you own the domain. For demonstration purposes, we use the Lightsail DNS zone. However, the steps might be similar for other DNS zones typically hosted by domain registrars.

Note

To learn more about how to create a Lightsail DNS zone for your domain, see [Creating a DNS zone to manage your domain's DNS records in Lightsail](#).

To add TXT records to your domain's DNS zone in Lightsail

1. On the Lightsail home page, choose the **Domains & DNS** tab.
2. Under the **DNS zones** section of the page, choose the DNS Zone for the domain that you specified in the Certbot certificate request.
3. In the DNS zone editor, choose **DNS records**.
4. Choose **Add record**.
5. In the **Record type** drop-down menu, choose **TXT record**.
6. Enter the values specified by the Let's Encrypt certificate request into the **Record name** and **Responds with** fields.

Note

The Lightsail console pre-populates the apex portion of your domain. For example, if you want to add the `_acme-challenge.example.com` subdomain, then you only have to enter `_acme-challenge` into the text box, and Lightsail adds the `.example.com` portion for you when you save the record.

7. Choose **Save**.
8. Repeat steps 4 through 7 to add the second set of TXT records specified by the Let's Encrypt certificate request.
9. Keep the Lightsail console browser window open—you return to it later in this tutorial. Continue to the [next section](#) of this tutorial.

Step 5: Confirm that the TXT records have propagated

Use the MxToolbox utility to confirm that the TXT records have propagated to the Internet's DNS. DNS record propagation might take a while depending on your DNS hosting provider, and the configured time to live (TTL) for your DNS records. It is important that you complete this step, and

confirm that your TXT records have propagated, before continuing your Certbot certificate request. Otherwise, your certificate request fails.

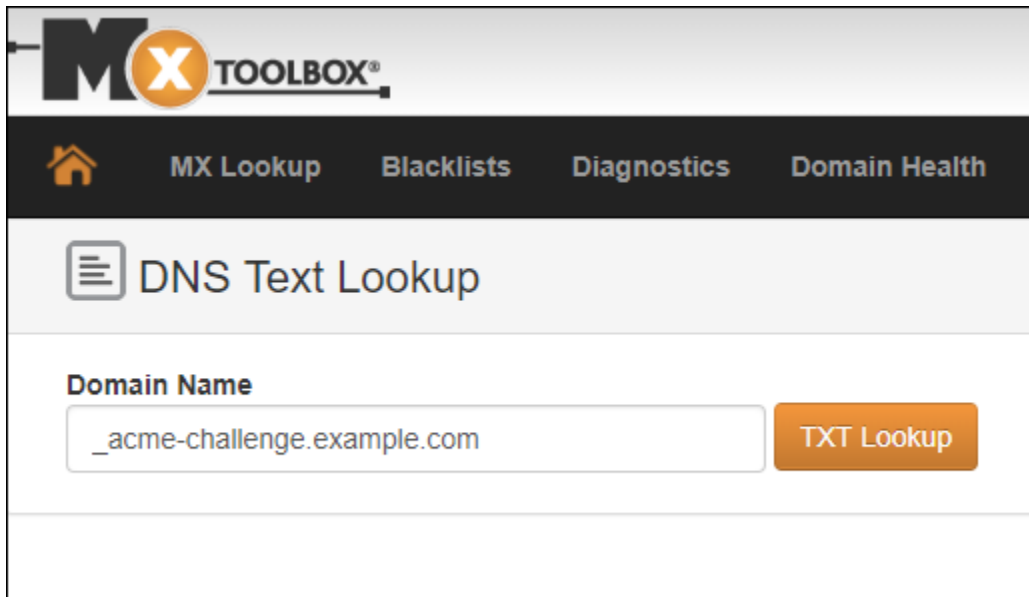
To confirm the TXT records have propagated to the Internet's DNS

1. Open a new browser window and go to <https://mxtoolbox.com/TXTLookup.aspx>.
2. Enter the following text into the text box. Be sure to replace *domain* with your domain.

`_acme-challenge.domain`

Example:

`_acme-challenge.example.com`



3. Choose **TXT Lookup** to run the check.
4. One of the following responses occurs:
 - If your TXT records have propagated to the Internet's DNS, you see a response similar to the one shown in the following screenshot. Close the browser window and continue to the [next section](#) of this tutorial.

txt:_acme-challenge.example.com [Find Problems](#) [txt](#)

Type	Domain Name	TTL	Record
TXT	_acme-challenge.example.com	60 sec	9vuaf232Bz0War8BUx3dTnBDpo6lm_4CDX4fpx4reoo
TXT	_acme-challenge.example.com	60 sec	BVkHW11a0Zhi2UB4BfoSmJV-B_fiSrwdaf8eBA30dU

	Test	Result
✓	DNS Record Published	DNS Record found

Your DNS hosting provider is "Amazon Route 53" [Need Bulk Dns Provider Data?](#)

[dns lookup](#)
[smtp diag](#)
[blacklist](#)
[http test](#)
[dns propagation](#)

Reported by on 10/8/2018 at 8:53:50 PM (UTC 0), [just for you.](#) [Transcript](#)

- If your TXT records have not propagated to the Internet's DNS, you see a **DNS Record not found** response. Confirm that you added the correct DNS records to your domains' DNS zone. If you added the correct records, wait a while longer to let your domain's DNS records propagate, and run the TXT lookup again.

Step 6: Complete the Let's Encrypt SSL certificate request

Go back to the Lightsail browser-based SSH session for your WordPress instance and complete the Let's Encrypt certificate request. Certbot saves your SSL certificate, chain, and key files to a specific directory on your WordPress instance.

To complete the Let's Encrypt SSL certificate request

1. In the Lightsail browser-based SSH session for your WordPress instance, press **Enter** to continue your Let's Encrypt SSL certificate request. If successful, a response similar to the one shown in the following screenshot appears:


```
-----
Please deploy a DNS TXT record under the name
_acme-challenge.example.com with the following value:

9vuaf232Bz0War8BUx3dTnBDpo6lm_4CDX4fpx4reoo

Before continuing, verify the record is deployed.
-----
Press Enter to Continue

-----
Please deploy a DNS TXT record under the name
_acme-challenge.example.com with the following value:

BVkHwll1a0Zhi2UB4BfoSmJV-B_fiSrWfdaf8eBA30dU

Before continuing, verify the record is deployed.
-----
Press Enter to Continue
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/example.com/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/example.com/privkey.pem
  Your cert will expire on 2019-01-06. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-interactively renew *all* of your certificates, run
  "certbot renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF: https://eff.org/donate-le

bitnami@ip-172-26-1-148:/$ █
```

The message confirms that your certificate, chain, and key files are stored in the `/etc/letsencrypt/live/domain/` directory. Make sure to replace *domain* with your domain, such as `/etc/letsencrypt/live/example.com/`.

2. Make note of the expiration date specified in the message. You use it to renew your certificate by that date.

```
IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/example.com/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/example.com/privkey.pem
  Your cert will expire on 2019-01-06. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-interactively renew *all* of your certificates, run
  "certbot renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF: https://eff.org/donate-le
```

3. Now that you have the Let's Encrypt SSL certificate, continue to the [next section](#) of this tutorial.

Step 7: Create links to the Let's Encrypt certificate files in the Apache server directory

Create links to the Let's Encrypt SSL certificate files in the Apache server directory on your WordPress instance. Also, back up your existing certificates, in case you need them later.

To create links to the Let's Encrypt certificate files in the Apache server directory

1. In the Lightsail browser-based SSH session for your WordPress instance, enter the following command to stop the underlying services:

```
sudo /opt/bitnami/ctlscript.sh stop
```

You should see a response similar to the following:

```
bitnami@ip-100-24-3-141:~$ sudo /opt/bitnami/ctlscript.sh stop
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
/opt/bitnami/php/scripts/ctl.sh : php-fpm stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
bitnami@ip-100-24-3-141:~$
```

2. Enter the following command to set an environment variable for your domain. You can more efficiently copy and paste commands to link the certificate files. Be sure to replace *domain* with the name of your registered domain name.

```
DOMAIN=domain
```

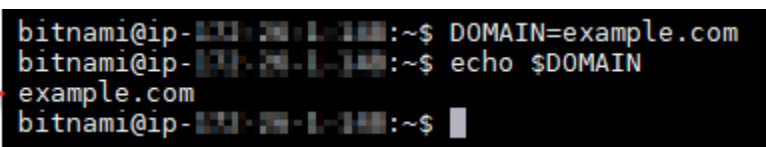
Example:

```
DOMAIN=example.com
```

3. Enter the following command to confirm the variables return the correct values:

```
echo $DOMAIN
```

You should see a result similar to the following:

A terminal window screenshot with a black background and white text. The prompt is 'bitnami@ip-100-20-1-100:~\$'. The first command is 'DOMAIN=example.com'. The second command is 'echo \$DOMAIN', and the output is 'example.com'. A red arrow points to the output line. The prompt is 'bitnami@ip-100-20-1-100:~\$' again.

```
bitnami@ip-100-20-1-100:~$ DOMAIN=example.com
bitnami@ip-100-20-1-100:~$ echo $DOMAIN
example.com
bitnami@ip-100-20-1-100:~$
```

4. Enter the following commands individually to rename your existing certificate files as backups. Refer to the **Important** block at the beginning of this tutorial for information about the different distributions and file structures.

- For Debian Linux distributions

Approach A (Bitnami installations using system packages):

```
sudo mv /opt/bitnami/apache2/conf/bitnami/certs/server.crt /opt/bitnami/apache2/conf/bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/apache2/conf/bitnami/certs/server.key /opt/bitnami/apache2/conf/bitnami/certs/server.key.old
```

Approach B (Self-contained Bitnami installations):

```
sudo mv /opt/bitnami/apache2/conf/server.crt /opt/bitnami/apache2/conf/server.crt.old
```

```
sudo mv /opt/bitnami/apache2/conf/server.key /opt/bitnami/apache2/conf/server.key.old
```

- For older instances that use the Ubuntu Linux distribution:

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.crt /opt/bitnami/apache/conf/bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.key /opt/bitnami/apache/conf/bitnami/certs/server.key.old
```

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.csr /opt/bitnami/apache/conf/bitnami/certs/server.csr.old
```

5. Enter the following commands individually to create links to your Let's Encrypt certificate files in the Apache directory. Refer to the **Important** block at the beginning of this tutorial for information about the different distributions and file structures.

- For Debian Linux distributions

Approach A (Bitnami installations using system packages):

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache2/conf/bitnami/certs/server.key
```

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache2/conf/bitnami/certs/server.crt
```

Approach B (Self-contained Bitnami installations):

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache2/conf/server.key
```

```
sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache2/conf/server.crt
```

- For older instances that use the Ubuntu Linux distribution:

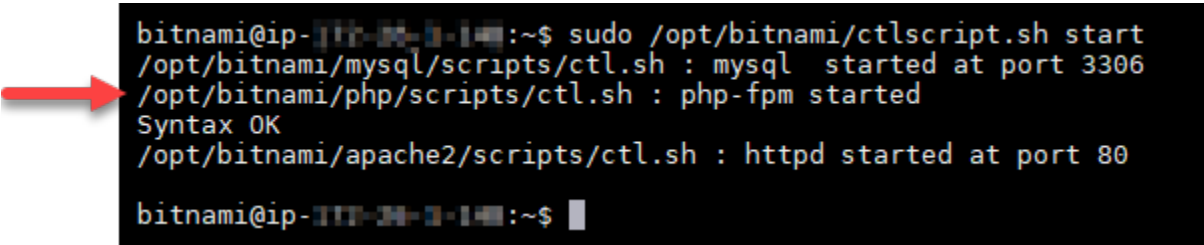
```
sudo ln -s /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache/conf/bitnami/certs/server.key
```

```
sudo ln -s /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache/conf/bitnami/certs/server.crt
```

6. Enter the following command to start the underlying services that you had stopped earlier:

```
sudo /opt/bitnami/ctlscript.sh start
```

You should see a result similar to the following:



```
bitnami@ip-10-10-10-10:~$ sudo /opt/bitnami/ctlscript.sh start
/opt/bitnami/mysql/scripts/ctl.sh : mysql started at port 3306
/opt/bitnami/php/scripts/ctl.sh : php-fpm started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80
bitnami@ip-10-10-10-10:~$
```

The SSL certificate files for your WordPress instance are now in the correct directory.

7. Continue to the [next section](#) of this tutorial.

Step 8: Integrate the SSL certificate with your WordPress site using the Really Simple SSL plug-in

Install the Really Simple SSL plug-in to your WordPress site, and use it to integrate the SSL certificate. Really Simple SSL also configures HTTP to HTTPS redirection to ensure that users who visit your site are always on the HTTPS connection.

To integrate the SSL certificate with your WordPress site using the Really Simple SSL plug-in

1. In the Lightsail browser-based SSH session for your WordPress instance, enter the following command to set your `wp-config.php` and `htaccess.conf` files to be writeable. The Really Simple SSL plug-in will write to the `wp-config.php` file to configure your certificates.

- For newer instances that use the Debian Linux distribution:

```
sudo chmod 666 /opt/bitnami/wordpress/wp-config.php && sudo chmod 666 /opt/bitnami/apache/conf/vhosts/htaccess/wordpress-htaccess.conf
```

- For older instances that use the Ubuntu Linux distribution:

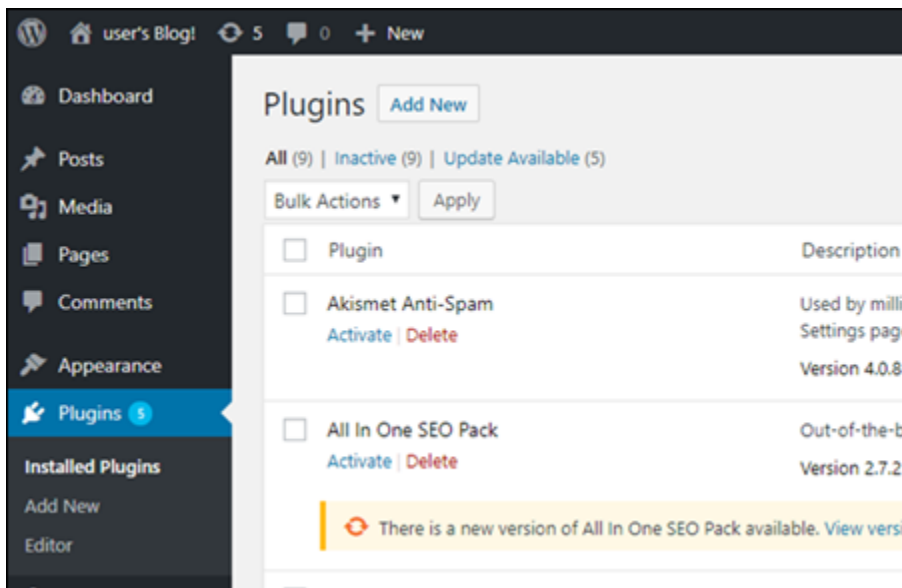
```
sudo chmod 666 /opt/bitnami/apps/wordpress/htdocs/wp-config.php && sudo chmod 666 /opt/bitnami/apps/wordpress/conf/htaccess.conf
```

2. Open a new browser window and sign in to the administration dashboard of your WordPress instance.

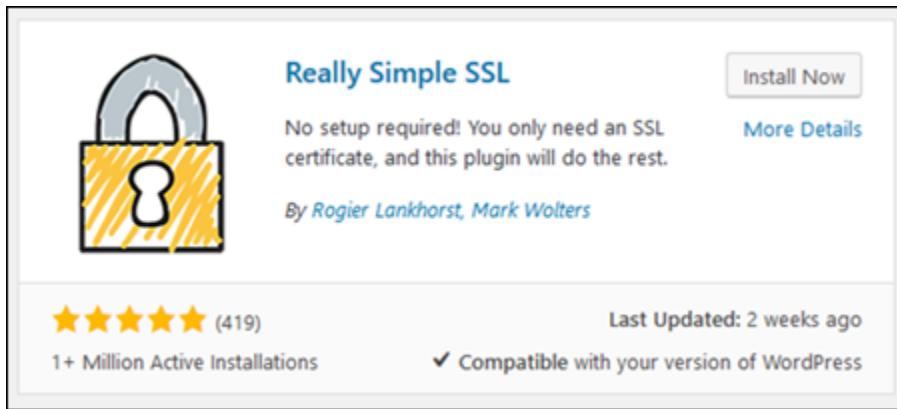
Note

For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

3. Choose **Plugins** from the left navigation pane.
4. Choose **Add New** from the top of the Plugins page.



5. Search for **Really Simple SSL**.
6. Choose **Install Now** next to the Really Simple SSL plug-in in the search results.



7. After it's done installing, choose **Activate**.
8. In the prompt that appears, choose **Go ahead, activate SSL!** You may be redirected to the sign in page for the administration dashboard of your WordPress instance.

Your WordPress instance is now configured to use SSL encryption. Additionally, your WordPress instance is now configured to automatically redirect connections from HTTP to HTTPS. When a visitor goes to `http://example.com`, they are automatically redirected to the encrypted HTTPS connection (i.e., `https://example.com`).

Step 9: Renew the Let's Encrypt certificates every 90 days

Let's Encrypt certificates are valid for 90 days. Certificates can be renewed 30 days before they expire. To renew the Let's Encrypt certificates, run the original command used to obtain them. Repeat the steps in the [Request a Let's Encrypt SSL wildcard certificate](#) section of this tutorial.

Networking tutorials for Amazon Lightsail

Use the following networking tutorials to explore Lightsail related topics, such as setting up Amazon VPC peering and configuring reverse DNS.

Topics

- [Configure IPv6 on cPanel instances in Lightsail](#)
- [Configure IPv6 on Debian 8 instances in Lightsail](#)
- [Configure IPv6 for GitLab instances in Lightsail](#)
- [Configure IPv6 on Nginx instances in Lightsail](#)
- [Configure IPv6 on Plesk instances in Lightsail](#)

- [Configure IPv6 for Ubuntu 16 instances in Lightsail](#)

Configure IPv6 on cPanel instances in Lightsail

All instances in Amazon Lightsail have a public and a private IPv4 address assigned to them by default. You can optionally enable IPv6 for your instances to have a public IPv6 address assigned to them. For more information, see [Amazon Lightsail IP addresses](#) and [Enable or disable IPv6](#).

After you enable IPv6 for an instance that uses the cPanel & WHM blueprint, you must perform an additional set of steps to make the instance aware of its IPv6 address. In this guide, we show you the additional steps that you must perform for cPanel & WHM instances.

Prerequisites

Complete the following prerequisites if you haven't already:

- Create an cPanel & WHM instance in Lightsail. For more information, see [Create an instance](#).
- Configure your cPanel & WHM instance. For more information, see [Quick start guide: cPanel & WHM on Amazon Lightsail](#).

Important

Make sure that all software updates and required system reboots are performed before continuing with the steps in this guide.

- Enable IPv6 for your cPanel & WHM instance. For more information, see [Enable or disable IPv6](#).

Note

New cPanel & WHM instances created on or after January 12, 2021, have IPv6 enabled by default when they are created in the Lightsail console. You must complete the following steps in this guide to configure IPv6 on your instance even if IPv6 was enabled by default when you created your instance.

Configure IPv6 on a cPanel & WHM instance

Complete the following procedure to configure IPv6 on a cPanel & WHM instance in Lightsail.

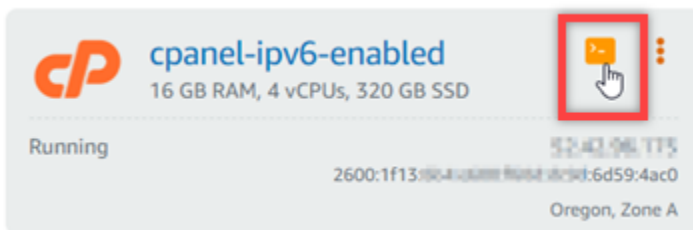
1. Sign in to the [Lightsail console](#).

- 2.

⚠ Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

In the **Instances** section of the Lightsail home page, locate the cPanel & WHM instance that you wish to configure, and choose the browser-based SSH client icon to connect to it using SSH.



3. After you're connected to your instance, enter the following command to open the `ifcfg-eth0` network interface configuration file using Nano.

```
sudo nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

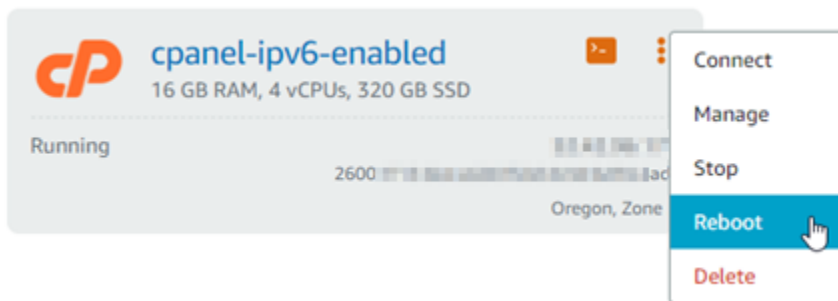
4. Add the following lines of text to the file if they are not already there.

```
IPV6INIT=yes  
IPV6_AUTOCONF=yes  
DHCPV6C=yes
```

The result should look like the following example.

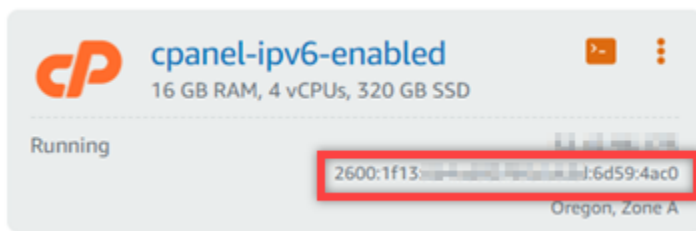
```
# Automatically generated by the vm import process
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
NAME=eth0
DEVICE=eth0
ONBOOT=yes
IPV6INIT=yes
IPV6_FAILURE_FATAL=no
DHCPV6C=yes
IPV6_AUTOCONF=yes
```

5. Press **CTRL+C** on your keyboard to exit the file.
6. Press **Y** when prompted to save the modified buffer, then press **Enter** to save to the existing file. This saves the edits you made to the `ifcfg-eth0` network interface configuration file.
7. Close the browser-based SSH window and toggle back to the Lightsail console.
8. In the **Instances** tab of the Lightsail home page, choose the actions menu (:) for the cPanel & WHM instance, and choose **Reboot**.

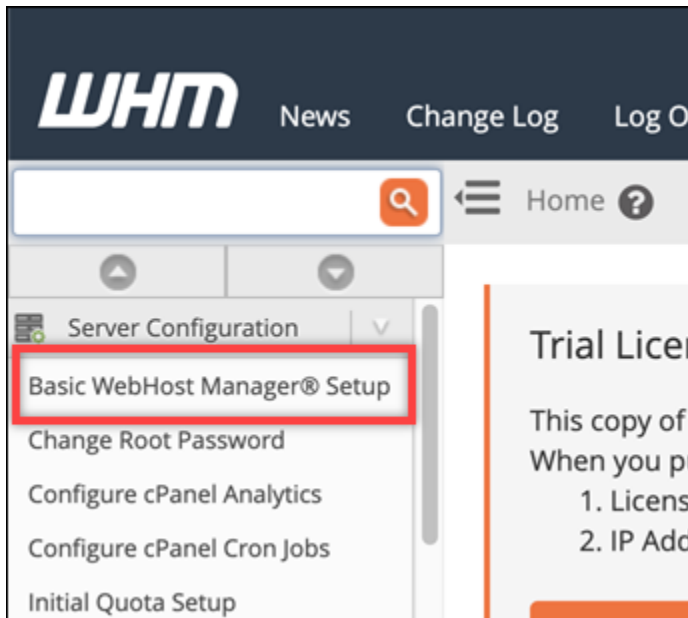


Wait a few minutes for your instance to be done rebooting before continuing to the next step.

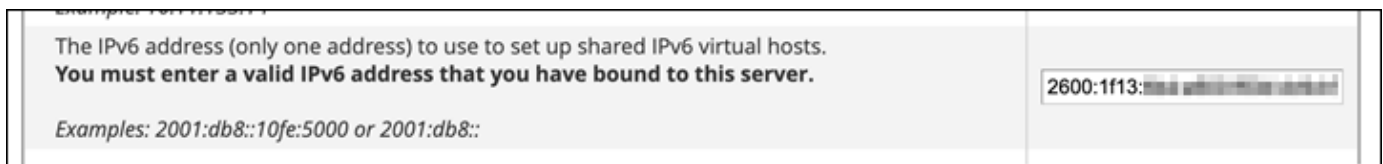
9. In the **Instances** tab of the Lightsail home page, make note of the IPv6 address assigned to your cPanel & WHM instance.



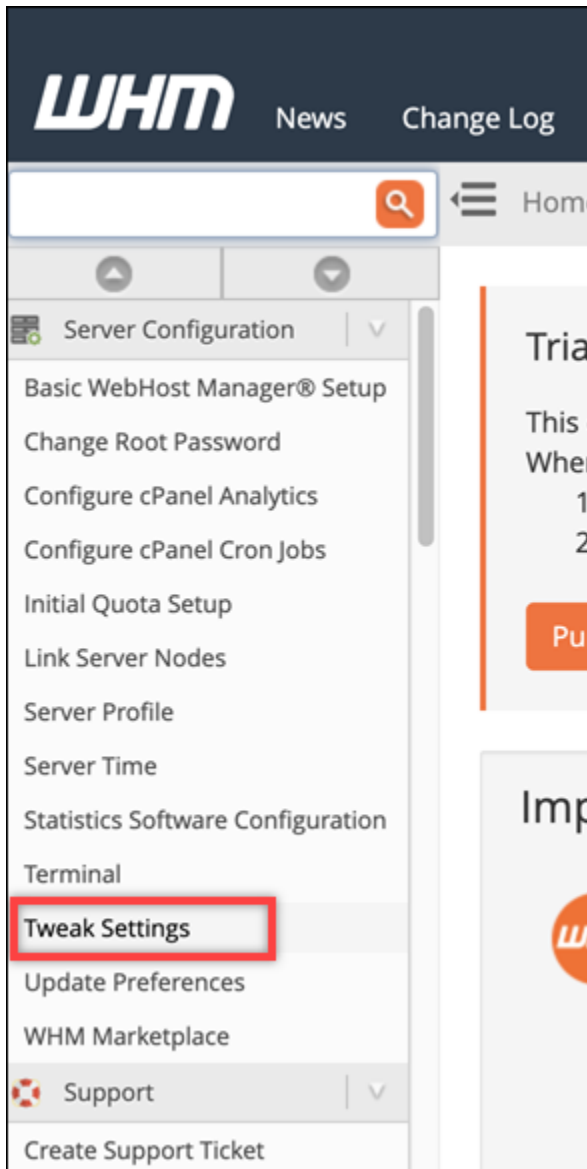
10. Open a new browser tab, and sign in to the Web Host Manager (WHM) of your cPanel & WHM instance.
11. In the left navigation pane of the WHM console, choose **Basic WebHost Manager Setup**.



12. In the **All** tab, find the text for the **IPv6 address to use**, then enter the IPv6 address assigned to your instance. You should have made note of the IPv6 address assigned to your instance from step 9 of this procedure.



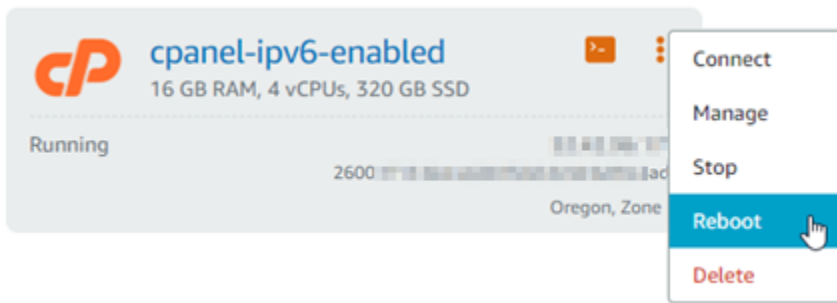
13. Scroll to the bottom for the page and choose **Save Changes**.
14. In the left navigation pane of the WHM console, choose **Tweak Settings**.



15. In the **All** tab, scroll down to find the **Listen on IPv6 Addresses** setting, and set it to **On**.

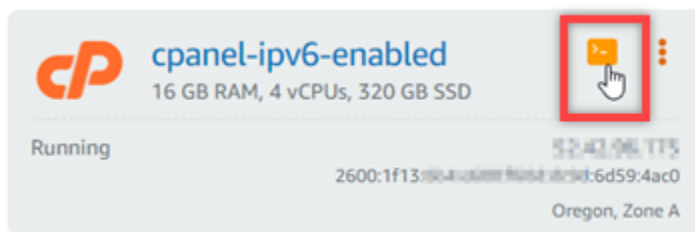


16. Scroll to the bottom for the page and choose **Save**.
17. Toggle back to the Lightsail console.
18. In the **Instances** tab of the Lightsail home page, choose the actions menu (:) for the cPanel & WHM instance, and choose **Reboot**.



Wait a few minutes for your instance to be done rebooting before continuing to the next step.

19. Choose the browser-based SSH client icon for the cPanel & WHM instance to connect to it using SSH.



20. After you're connected to your instance, enter the following command to view the IP addresses configured on your instance, and confirm that it is now recognizing its assigned IPv6 address.

```
ip addr
```

You will see a response similar to the following example. If your instance does recognize its IPv6 address, then you will see it listed in the response with a label of **scope global** as shown in this example.

```
[centos@52-42-94-175 ~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
     valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
     valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
   link/ether 02:9b:51:92:50:45 brd ff:ff:ff:ff:ff:ff
   inet 172.31.0.1/20 brd 172.31.255.255 scope global dynamic eth0
     valid_lft 2301sec preferred_lft 2301sec
   inet6 2600:1f13:804::6d59:4ac0/128 scope global dynamic
     valid_lft 112sec preferred_lft 112sec
   inet6 fe80::9045:5045:5045:5045/64 scope link
     valid_lft forever preferred_lft forever
```

21. Enter the following command to confirm that your instance is able to ping an IPv6 address.

```
ping6 ipv6.google.com -c 6
```

The result should look like the following example, which confirms that your instance is able to ping IPv6 addresses.

```
[centos@32-42-74-173 ~]$ ping6 ipv6.google.com
PING ipv6.google.com(sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e)) 56 data bytes
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=1 ttl=103 time=7.66 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=2 ttl=103 time=7.70 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=3 ttl=103 time=7.68 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=4 ttl=103 time=7.69 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=5 ttl=103 time=7.70 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=6 ttl=103 time=7.68 ms
^C
--- ipv6.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 7.667/7.690/7.702/0.052 ms
```

Configure IPv6 on Debian 8 instances in Lightsail

All instances in Amazon Lightsail have a public and a private IPv4 address assigned to them by default. You can optionally enable IPv6 for your instances to have a public IPv6 address assigned to them. For more information, see [Amazon Lightsail IP addresses](#) and [Enable or disable IPv6](#).

After you enable IPv6 for an instance that uses the Debian 8 blueprint, you must perform an additional set of steps to make the instance aware of its IPv6 address. In this guide, we show you the additional steps that you must perform for Debian 8 instances.

Prerequisites

Complete the following prerequisites if you haven't already:

- Create a Debian 8 instance in Lightsail. For more information, see [Create an instance](#).
- Enable IPv6 for your Debian 8 instance. For more information, see [Enable or disable IPv6](#).

Note

New Debian instances created on or after January 12, 2021, have IPv6 enabled by default when they are created in the Lightsail console. You must complete the following steps in this guide to configure IPv6 on your instance even if IPv6 was enabled by default when you created your instance.

Configure IPv6 on a Debian 8 instance

Complete the following procedure to configure IPv6 on a Debian 8 instance in Lightsail.

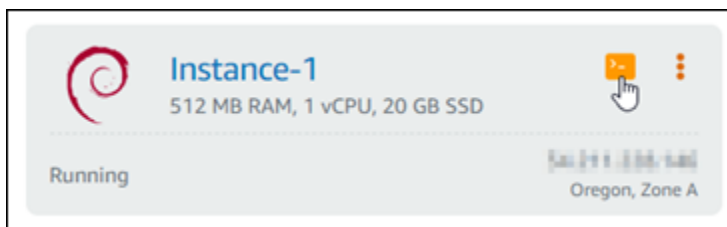
1. Sign in to the [Lightsail console](#).

- 2.

Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

In the **Instances** section of the Lightsail home page, locate the Debian 8 instance that you wish to configure, and choose the browser-based SSH client icon to connect to it using SSH.



3. After you're connected to your instance, enter the following command to view the IP addresses configured on your instance.

```
ip addr
```

You will see a response similar to one of the following examples:

- If your instance *does not* recognize its IPv6 address, then you will not see it listed in the response. You should continue to complete steps 4 through 9 of this procedure.

```
admin@ip-172-31-0-228:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:9c:ad:11:00:00:00:00:00:00:ff:ff
    inet 172.31.0.228/20 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::209c:ad11:0000:0000:3df7/64 scope link
        valid_lft forever preferred_lft forever
```

- If your instance does recognize its IPv6 address, then you will see it listed in the response with a scope `global` as shown in this example. You should stop here; you do not need to complete steps 4 through 9 of this procedure because your instance is already configure to recognize its IPv6 address.

```
admin@ip-172-31-0-228:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:9c:ad:11:00:00:00:00:00:00:ff:ff
    inet 172.31.0.228/20 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 2600:1f13:1000:1000:1000:1000:f383:3212/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::209c:ad11:0000:0000:3df7/64 scope link
        valid_lft forever preferred_lft forever
```

4. Enter the following command to open the `interfaces` configuration file using Nano.

```
sudo nano /etc/network/interfaces
```

5. Add the following line of text to the end of the file.

```
iface eth0 inet6 dhcp
```

The file will look like the following when done:


```
GNU nano 2.2.6 File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp

iface eth1 inet dhcp
iface eth2 inet dhcp
iface eth3 inet dhcp
iface eth4 inet dhcp
iface eth5 inet dhcp
iface eth6 inet dhcp
iface eth7 inet dhcp
iface eth0 inet6 dhcp
```

6. Press the **Ctrl+Esc** keys to exit Nano.
7. Press **Y** when asked if you would like to save the modified buffer, then press **Enter** to save to the existing interfaces configuration file.
8. Enter the following command to restart the networking service of your instance.

```
sudo systemctl restart networking
```

You might need to wait a few more minutes to allow your instance to recognize its IPv6 address after you restart the networking service of your instance.

9. Enter the following command to view the IP addresses configured on your instance, and confirm that it is now recognizing its assigned IPv6 address.

```
ip addr
```

You will see a response similar to the following example. If your instance does recognize its IPv6 address, then you will see it listed in the response with a label of `scope global` as shown in this example.

```
admin@ip-172-31-1-228:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:9c:84:1f:0a:ff brd ff:ff:ff:ff:ff:ff
    inet 172.31.1.228/24 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 2600:1f13:1000:1000:1000:1000:f383:3212/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::1000:1000:1000:1000:3df7/64 scope link
        valid_lft forever preferred_lft forever
```

Configure IPv6 for GitLab instances in Lightsail

All instances in Amazon Lightsail have a public and a private IPv4 address assigned to them by default. You can optionally enable IPv6 for your instances to have a public IPv6 address assigned to them. For more information, see [Amazon Lightsail IP addresses](#) and [Enable or disable IPv6](#).

After you enable IPv6 for an instance that uses the GitLab blueprint, you must perform an additional set of steps to make the instance aware of its IPv6 address. In this guide, we show you the additional steps that you must perform for GitLab instances.

Prerequisites

Complete the following prerequisites if you haven't already:

- Create a GitLab instance in Lightsail. For more information, see [Create an instance](#).
- Enable IPv6 for your GitLab instance. For more information, see [Enable or disable IPv6](#).

Note

New GitLab instances created on or after January 12, 2021, have IPv6 enabled by default when they are created in the Lightsail console. You must complete the following steps in this guide to configure IPv6 on your instance even if IPv6 was enabled by default when you created your instance.

Configure IPv6 on a GitLab instance

Complete the following procedure to configure IPv6 on a GitLab instance in Lightsail.

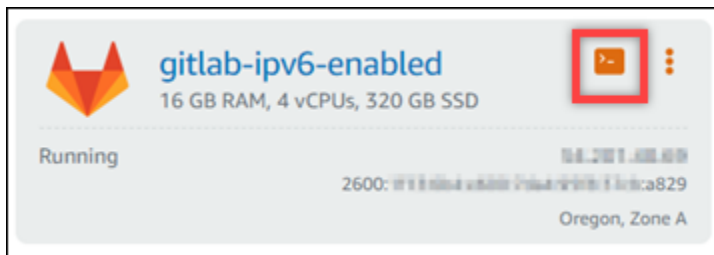
1. Sign in to the [Lightsail console](#).

2.

⚠ Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

In the **Instances** section of the Lightsail home page, locate the GitLab instance that you wish to configure, and choose the browser-based SSH client icon to connect to it using SSH.



3. After you're connected to your instance, enter the following command to view the IP addresses configured on your instance.

```
ip addr
```

You will see a response similar to one of the following examples:

- If your instance does not recognize its IPv6 address, then you will not see it listed in the response. You should continue to complete steps 4 through 9 of this procedure.

```
admin@ip-172.31.0.1:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:9c:ad:00:00:00:00:00:00:00:00:00:ff:ff
    inet 172.31.0.1/16 brd 172.31.255.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::209c:adff:fe00:0000:3df7/64 scope link
        valid_lft forever preferred_lft forever
```

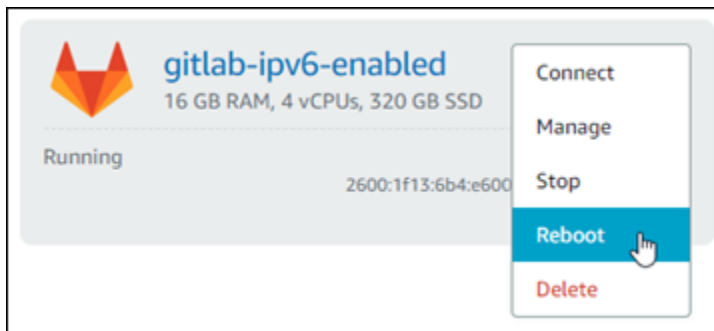
- If your instance does recognize its IPv6 address, then you will see it listed in the response with a scope `global` as shown in this example. You should stop here; you do not need to complete steps 4 through 9 of this procedure because your instance is already configured to recognize its IPv6 address.

```

admin@ip-172-31-4-200:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:9c:84:11:00:00:00:00:00:00:00:00:ff:ff
    inet 172.31.4.200/20 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 2600:1f13:6b4:e600::f383:3212/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::84:11:00:00:00:00:00:00:3df7/64 scope link
        valid_lft forever preferred_lft forever

```

4. Toggle back to the Lightsail console.
5. In the **Instances** tab of the Lightsail home page, choose the actions menu (:) for the GitLab instance, and choose **Reboot**.



Wait a few minutes for your instance to be done rebooting before continuing to the next step.

6. Toggle back to the SSH session of your GitLab instance.
7. Enter the following command to view the IP addresses configured on your instance, and confirm that it is now recognizing its assigned IPv6 address.

```
ip addr
```

You will see a response similar to the following example. If your instance does recognize its IPv6 address, then you will see it listed in the response with a label of `scope global` as shown in this example.

```
admin@ip-172-31-1-23:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:9c:84:1f:0a:ff brd ff:ff:ff:ff:ff:ff
    inet 172.31.1.23/24 brd 172.31.1.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 2600:1f13:1000:1000:1000:1000:f383:3212/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::1000:1000:1000:1000:3df7/64 scope link
        valid_lft forever preferred_lft forever
```

Configure IPv6 on Nginx instances in Lightsail

All instances in Amazon Lightsail have a public and a private IPv4 address assigned to them by default. You can optionally enable IPv6 for your instances to have a public IPv6 address assigned to them. For more information, see [Amazon Lightsail IP addresses](#) and [Enable or disable IPv6](#).

After you enable IPv6 for an instance that uses the Nginx blueprint, you must perform an additional set of steps to make the instance aware of its IPv6 address. In this guide, we show you the additional steps that you must perform for Nginx instances.

Prerequisites

Complete the following prerequisites if you haven't already:

- Create an Nginx instance in Lightsail. For more information, see [Create an instance](#).
- Enable IPv6 for your Nginx instance. For more information, see [Enable or disable IPv6](#).

Note

New Nginx instances created on or after January 12, 2021, have IPv6 enabled by default when they are created in the Lightsail console. You must complete the following steps in this guide to configure IPv6 on your instance even if IPv6 was enabled by default when you created your instance.

Configure IPv6 on a Nginx instance

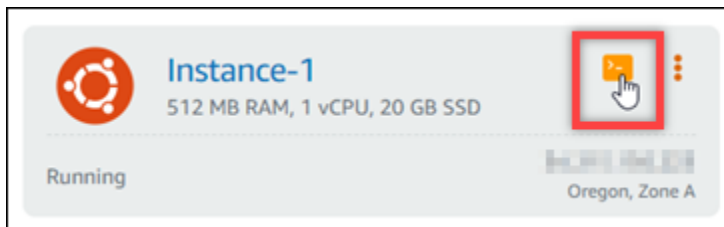
Complete the following procedure to configure IPv6 on a Nginx instance in Lightsail.

1. Sign in to the [Lightsail console](#).
- 2.

⚠ Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

In the **Instances** section of the Lightsail home page, locate the Ubuntu 16 instance that you wish to configure, and choose the browser-based SSH client icon to connect to it using SSH.



3. After you're connected to your instance, enter the following command to determine if your instance is listening to IPv6 requests over port 80. Be sure to replace *<IPv6Address>* with the IPv6 address assigned to your instance.

```
curl -g -6 'http://[<IPv6Address>]'
```

Example:

```
curl -g -6 'http://[2001:0db8:85a3:0000:0000:8a2e:0370:7334]'
```

You will see a response similar to one of the following examples:

- If your instance is not listening to IPv6 requests over port 80, then you will see a response with a **Failed to connect** error message. You should continue to complete steps 4 through 9 of this procedure.

```
bitnami@ip-172-31-0-104:~$ curl -g -6 'http://[2600:1f13:8000:0000:0000:985b:25d9]:80'
curl: (7) Failed to connect to 2600:1f13:8000:0000:0000:985b:25d9 port 80: Connection refused
```

- If your instance is listening to IPv6 requests over port 80, then you will see a response with the HTML code of the home page of your instance as shown in the following example. You should stop here; you do not need to complete steps 4 through 9 of this procedure because your instance is already configured for IPv6.

```

bitnami@ip-10.0.0.10:~$ curl -g -6 'http://[2600:1202:6000:1000:1000:985b:25d9]:80'
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Bitnami NGINX Open Source</title>
    <meta name="description" content="Bitnami: Open Source. Simplified.">
    <meta name="author" content="Bitnami">
    <link rel="stylesheet" media="screen" href="//unpkg.com/@bitnami/hex/dist/hex.min.css">
  </head>
  <body>
    <main class="margin-t-huge">
      <section aria-labelledby="installation-title" aria-describedby="installation-desc" class="container container-tiny margin-b-gi
      <h1 id="installation-title">Congratulations!</h1>
      <div aria-hidden="true" style="height: 4px; width: 100%;" class="gradient-135-brand"></div>
      <p id="installation-desc" class="type-big">You are now running <strong>Bitnami NGINX Open Source 1.18.0</strong> in the Clou
      </section>
      <section aria-labelledby="links-title" aria-describedby="links-desc" class="bg-light padding-v-bigger margin-v-enormous">
        <div class="container container-tiny">
          <div class="row row-collapse-b-tablet align-center ">
            <div class="col-6">
              <h3 id="links-title" class="margin-t-reset">Useful Links</h3>
              <p id="links-desc" class="margin-b-reset">The following links will help you to understand better how to get started an
            </div>
          </div>
        </div>
      </section>
    </main>
  </body>
</html>
unched.</p>

```

4. Enter the following command to open the `nginx.conf` configuration file using Vim.

```
sudo vim /opt/bitnami/nginx/conf/nginx.conf
```

5. Press `I` to enter insert mode in Vim.
6. Add the following text below the `listen 80;` text that is already in the file. You might need to scroll down in Vim to see the section where you need to add the text.

```
listen [::]:80;
```

The file will look like the following when done:

```

client_max_body_size 80m;
server_tokens off;

include "/opt/bitnami/nginx/conf/server_blocks/*.conf";

# HTTP Server
server {
    # Port to listen on, can also be set in IP:PORT format
    listen 80;
    listen [::]:80;

    include "/opt/bitnami/nginx/conf/bitnami/*.conf";

    location /status {
        stub_status on;
        access_log off;
        allow 127.0.0.1;
        deny all;
    }
}

```

7. Press the **Esc** key to exit insert mode in Vim, then type `:wq!` and press **Enter** to save your edits (write) and quit Vim.
8. Enter the following command to restart the services of your instance.

```
sudo /opt/bitnami/ctlscript.sh restart
```

9. Enter the following command to determine if your instance is listening to IPv6 requests over port 80. Be sure to replace `<IPv6Address>` with the IPv6 address assigned to your instance.

```
curl -g -6 'http://[<IPv6Address>]'
```

Example:

```
curl -g -6 'http://[2001:0db8:85a3:0000:0000:8a2e:0370:7334]'
```

You will see a response similar to the following example. If your instance is listening to IPv6 requests over port 80, then you will see a response with the HTML code of the home page of your instance.

```
bitnami@ip-...:~$ curl -g -6 'http://[2600:1f19:1000:1000:1000:1000:985b:25d9]:80'
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Bitnami NGINX Open Source</title>
    <meta name="description" content="Bitnami: Open Source. Simplified.">
    <meta name="author" content="Bitnami">
    <link rel="stylesheet" media="screen" href="//unpkg.com/@bitnami/hex/dist/hex.min.css">
  </head>
  <body>
    <main class="margin-t-huge">
      <section aria-labelledby="installation-title" aria-describedby="installation-desc" class="container container-tiny margin-b-gi
      <h1 id="installation-title">Congratulations!</h1>
      <div aria-hidden="true" style="height: 4px; width: 100%;" class="gradient-135-brand"></div>
      <p id="installation-desc" class="type-big">You are now running <strong>Bitnami NGINX Open Source 1.18.0</strong> in the Clou
      </section>
      <section aria-labelledby="links-title" aria-describedby="links-desc" class="bg-light padding-v-bigger margin-v-enormous">
        <div class="container container-tiny">
          <div class="row row-collapse-b-tablet align-center ">
            <div class="col-6">
              <h3 id="links-title" class="margin-t-reset">Useful Links</h3>
              <p id="links-desc" class="margin-b-reset">The following links will help you to understand better how to get started an
            </div>
          </div>
        </div>
      </section>
    </main>
  </body>
</html>
```

Configure IPv6 on Plesk instances in Lightsail

All instances in Amazon Lightsail have a public and a private IPv4 address assigned to them by default. You can optionally enable IPv6 for your instances to have a public IPv6 address assigned to them. For more information, see [Amazon Lightsail IP addresses](#) and [Enable or disable IPv6](#).

After you enable IPv6 for an instance that uses the Plesk blueprint, you must perform an additional set of steps to make the instance aware of its IPv6 address. In this guide, we show you the additional steps that you must perform for Plesk instances.

Prerequisites

Complete the following prerequisites if you haven't already:

- Create an Plesk instance in Lightsail. For more information, see [Create an instance](#).
- Enable IPv6 for your Plesk instance. For more information, see [Enable or disable IPv6](#).

Note

New Plesk instances created on or after January 12, 2021, have IPv6 enabled by default when they are created in the Lightsail console. You must complete the following steps in this guide to configure IPv6 on your instance even if IPv6 was enabled by default when you created your instance.

Configure IPv6 on a Plesk instance

Complete the following procedure to configure IPv6 on a Plesk instance in Lightsail.

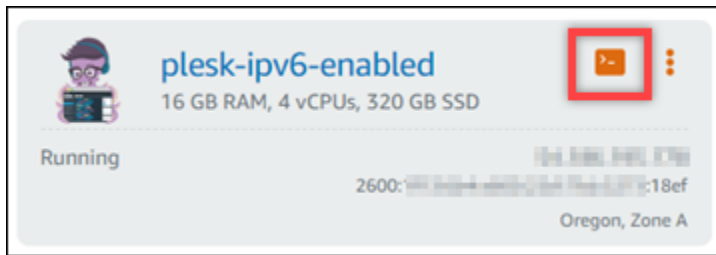
1. Sign in to the [Lightsail console](#).

- 2.

Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

In the **Instances** section of the Lightsail home page, locate the Plesk instance that you wish to configure, and choose the browser-based SSH client icon to connect to it using SSH.



3. After you're connected to your instance, enter the following command to view the IP addresses configured on your instance.

```
ip addr
```

You will see a response similar to one of the following examples:

- If your instance does not recognize its IPv6 address, then you will not see it listed in the response. You should continue to complete steps 4 through 7 of this procedure.

```
admin@ip-172-31-0-253:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 02:9c:ad:11:00:00:00:00:00:00:00:00:ff:ff
   inet 172.31.0.253/20 brd 172.31.0.255 scope global eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::209c:ad11:0000:0000:3df7/64 scope link
       valid_lft forever preferred_lft forever
```

- If your instance does recognize its IPv6 address, then you will see it listed in the response with a scope `global` as shown in this example. You should stop here; you do not need to complete steps 4 through 7 of this procedure because your instance is already configured to recognize its IPv6 address.

```
admin@ip-172-31-0-253:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 02:9c:ad:11:00:00:00:00:00:00:00:00:ff:ff
   inet 172.31.0.253/20 brd 172.31.0.255 scope global eth0
       valid_lft forever preferred_lft forever
   inet6 2600:1f13:1200:1000::f383:3212/64 scope global
       valid_lft forever preferred_lft forever
   inet6 fe80::209c:ad11:0000:0000:3df7/64 scope link
       valid_lft forever preferred_lft forever
```

4. Toggle back to the Lightsail console.

After you enable IPv6 for an instance that uses the Ubuntu 16 blueprint, you must perform an additional set of steps to make the instance aware of its IPv6 address. In this guide, we show you the additional steps that you must perform for Ubuntu 16 instances.

Prerequisites

Complete the following prerequisites if you haven't already:

- Create a Ubuntu 16 instance in Lightsail. For more information, see [Create an instance](#).
- Enable IPv6 for your Ubuntu 16 instance. For more information, see [Enable or disable IPv6](#).

Note

New Ubuntu instances created on or after January 12, 2021, have IPv6 enabled by default when they are created in the Lightsail console. You must complete the following steps in this guide to configure IPv6 on your instance even if IPv6 was enabled by default when you created your instance.

Configure IPv6 on a Ubuntu 16 instance

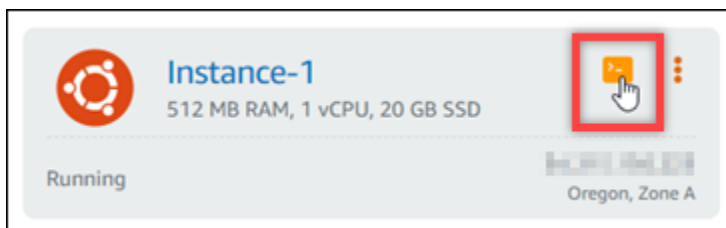
Complete the following procedure to configure IPv6 on a Ubuntu 16 instance in Lightsail.

1. Sign in to the [Lightsail console](#).
- 2.

Important

The Lightsail browser-based SSH/RDP clients only accept IPv4 traffic. Use a third-party client to SSH or RDP into your instance over IPv6. For more information, see [Connect to your instances](#).

In the **Instances** section of the Lightsail home page, locate the Ubuntu 16 instance that you wish to configure, and choose the browser-based SSH client icon to connect to it using SSH.



3. After you're connected to your instance, enter the following command to view the IP addresses configured on your instance.

```
ip addr
```

You will see a response similar to one of the following examples:

- If your instance *does not* recognize its IPv6 address, then you will not see it listed in the response. You should continue to complete steps 4 through 9 of this procedure.

```
ubuntu@ip-172-36-4-4:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 02:af:1e:00:16:bf brd ff:ff:ff:ff:ff:ff
   inet 172.36.4.4/20 brd 172.36.15.255 scope global eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::af:1e1f:fed3:16bf/64 scope link
       valid_lft forever preferred_lft forever
```

- If your instance does recognize its IPv6 address, then you will see it listed in the response with a scope `global` as shown in this example. You should stop here; you do not need to complete steps 4 through 9 of this procedure because your instance is already configured to recognize its IPv6 address.

```
ubuntu@ip-172-36-4-4:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 02:af:1e:00:16:bf brd ff:ff:ff:ff:ff:ff
   inet 172.36.4.4/20 brd 172.36.15.255 scope global eth0
       valid_lft forever preferred_lft forever
   inet6 2600:1f13:edc:ed2c:91e2/128 scope global
       valid_lft forever preferred_lft forever
   inet6 fe80::af:1e1f:fed3:16bf/64 scope link
       valid_lft forever preferred_lft forever
```

4. Enter the following command to open the interfaces configuration file using Vim.

```
sudo vim /etc/network/interfaces
```

5. Press `I` to enter insert mode in Vim.
6. Add the following line of text to the end of the file.

```
iface eth0 inet6 dhcp
```

The file will look like the following when done:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# Source interfaces
# Please check /etc/network/interfaces.d before changing this file
# as interfaces may have been defined in /etc/network/interfaces.d
# See LP: #1262951
source /etc/network/interfaces.d/*.cfg

iface eth0 inet6 dhcp
```

7. Press the **Esc** key to exit insert mode in Vim, then type `:wq!` and press **Enter** to save your edits (write) and quit Vim.
8. Enter the following command to restart the networking service of your instance.

```
sudo service networking restart
```

You might need to wait a few more minutes to allow your instance to recognize its IPv6 address after you restart the networking service of your instance.

9. Enter the following command to view the IP addresses configured on your instance, and confirm that it is now recognizing its assigned IPv6 address.

```
ip addr
```

You will see a response similar to the following example. If your instance does recognize its IPv6 address, then you will see it listed in the response with a label of `scope global` as shown in this example.

```
ubuntu@ip-172-31-4-1:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:af:fe:d3:16:bf brd ff:ff:ff:ff:ff:ff
    inet 172.31.4.200/20 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 2600:1f13:404:4000:2e77:700c:ed2c:91e2/128 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::af:fe:d3:16bf/64 scope link
        valid_lft forever preferred_lft forever
```

Work with Amazon Lightsail

Use the following tutorials to learn about various tasks you can complete in Lightsail. For example, you can create a HAR file for troubleshooting, launch and configure a LAMP instance, or migrate your MySQL database.

Topics

- [Work with the AWS Command Line Interface in Lightsail](#)
- [Create an access key to use the Lightsail API or the AWS Command Line Interface](#)
- [AWS CloudShell in Lightsail](#)
- [Log Lightsail API calls with AWS CloudTrail](#)
- [Tutorial: Connect a Lightsail LAMP instance to an Aurora database](#)
- [Tutorial: How to create a HAR file](#)
- [Force stop your Lightsail instance](#)
- [Tutorial: Install Prometheus on a Linux-based Lightsail instance](#)
- [Tutorial: Launch and configure a Lightsail LAMP instance](#)
- [Tutorial: Launch and configure a Windows Server 2016 instance](#)
- [Learn more about Amazon Lightsail](#)
- [Tutorial: Migrate data from a MySQL 5.6 database to a newer database version](#)
- [Set up and configure Plesk in Lightsail](#)
- [Tutorial: Use a Lightsail bucket with a content delivery network distribution](#)
- [Use Lightsail with other AWS services](#)
- [Create Lightsail resources with AWS CloudFormation](#)

Work with the AWS Command Line Interface in Lightsail

The AWS Command Line Interface (AWS CLI) is a tool that allows advanced users and developers to control the Amazon Lightsail service by typing commands in the terminal (on Linux and Unix) or Command Prompt (on Windows). You can also control Lightsail using the Lightsail console, a graphical user interface, and the Lightsail application program interface (API).

In Lightsail, you can install the AWS CLI on your local desktop or install it on your Lightsail instance.

For more information about the AWS CLI, see [AWS Command Line Interface User Guide](#). You can find the Amazon Lightsail commands in the [AWS CLI Command Reference](#).

- To install the AWS CLI on your local desktop, see [Installing the AWS CLI](#) in the AWS Command Line Interface documentation.
- To install the AWS CLI on your Ubuntu-based Lightsail instance, connect to your instance, and type `sudo apt-get -y install awscli`.

Note

The AWS CLI should already be installed on the Amazon Linux Lightsail instance. If you need to reinstall it, connect to your instance, and type `sudo yum install aws-cli`.

After you install the AWS CLI, you need to obtain access keys and then configure the AWS CLI to use them. For more information, see [Create an access key to use the Lightsail API or the AWS Command Line Interface](#).


Create an access key to use the Lightsail API or the AWS Command Line Interface

To use the Lightsail API or the AWS Command Line Interface (AWS CLI), you need to create a new access key. The access key consists of an **Access Key ID** and a **Secret Access Key**. Use the following procedures to create the key and configure the AWS CLI to make calls to the Lightsail API.

Step 1: Create a new access key

You can create a new access key in the AWS Identity and Access Management (IAM) console.

1. Sign in to the [the IAM console](#).
2. Choose the name of the user for which you want to create an access key. The user you choose should have full access or specific access to Lightsail actions.
3. Choose the **Security credentials** tabs.
4. Choose **Create access key** under the **Access keys** section of the page.

 **Note**

You can have a maximum of two access keys (active or inactive) at a time per user. If you already have two access keys, then you must delete one of them before creating a new one. Make sure that an access key is not actively in use before deleting it.

5. Make note of the **Access key ID** and **Secret access key** listed. Choose **Show** under the **Secret access key** column to see your **Secret access key**.

You can copy them from this screen or choose **Download Key File** to download a .csv file containing the access key ID and secret access key.

 **Important**

Keep your access keys in a safe place. You should name the file something like `MyLightsailKeys.csv` so that you don't struggle to find them later. If you've downloaded the CSV file from the IAM console, you should delete it after you've completed step 2. You can create a new access keys later if you need to.

Step 2: Configure the AWS CLI

If you haven't installed the AWS CLI, you can do that now. See [Installing the AWS Command Line Interface](#). After you install the AWS CLI, you need to configure it so you can use it.

1. Open a terminal window or command prompt.
2. Type `aws configure`.
3. Paste your **AWS Access Key ID** from the .csv file you created in the previous step.
4. Paste your **AWS Secret Access Key** when prompted.
5. Enter the AWS Region where your resources are located. For example, if your resources are primarily in Ohio, choose `us-east-2` when prompted for the **Default region name**.

For more information about using the AWS CLI `--region` option, see [General Options](#) in the *AWS CLI Reference*.

6. Choose a **Default output format**, such as `json`.

Next steps

- [Install the SDK](#)
- [Configure the AWS Command Line Interface to work with Amazon Lightsail](#)
- [Read the API docs](#)

AWS CloudShell in Lightsail

AWS CloudShell is a browser-based, pre-authenticated shell that you can launch directly from the Amazon Lightsail console. Use CloudShell to manage your Lightsail resources from the command line interface. You can run AWS Command Line Interface (AWS CLI) commands using your preferred shell, such as Bash, PowerShell, or Z shell. You can do this without downloading or installing command line tools. When you launch CloudShell, a [compute environment](#) that's based on Amazon Linux 2 is created. Within this environment, you can access an extensive range of pre-installed development tools, such as the AWS CLI. For a complete list of pre-installed tools, see [Pre-installed software](#) in the *CloudShell User Guide*.

Persistent storage

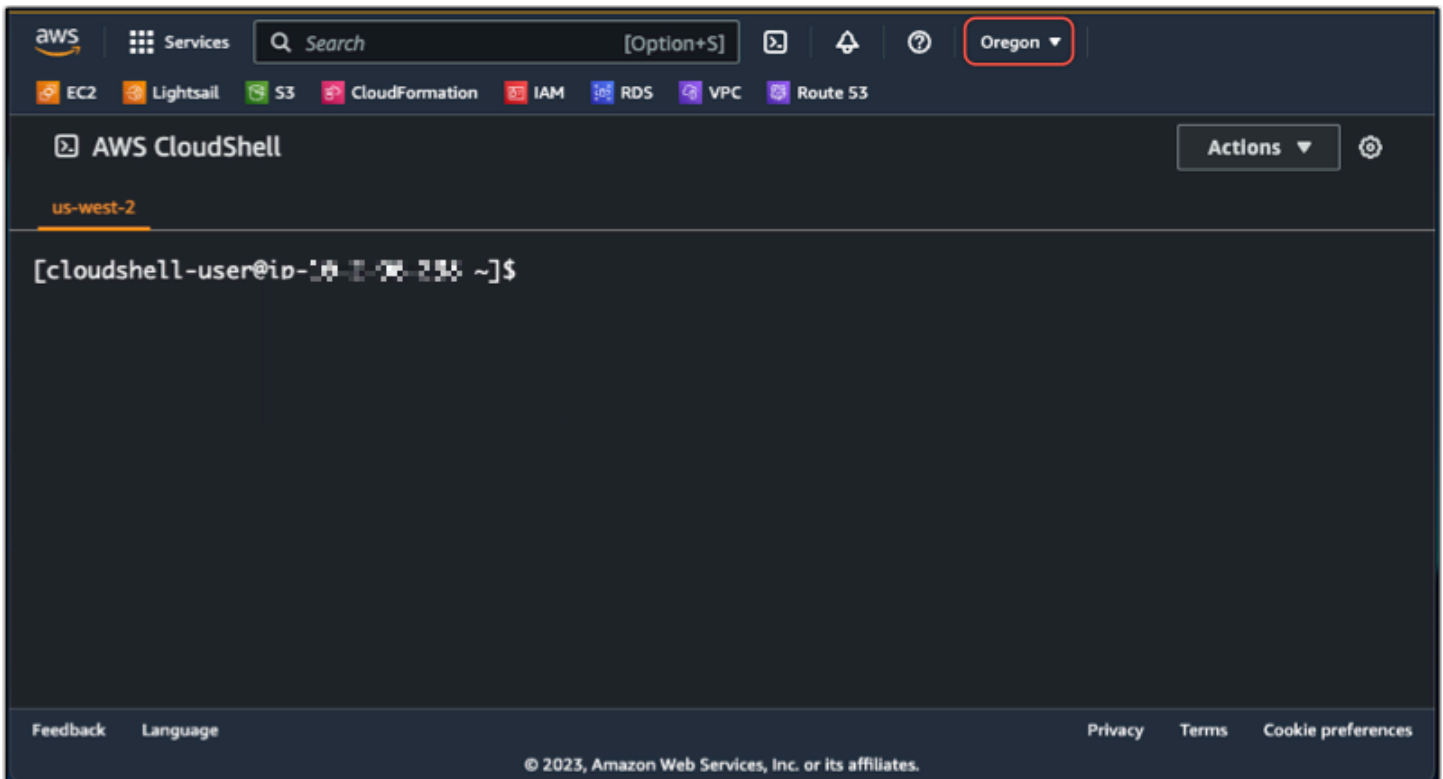
With AWS CloudShell, you can use up to 1 GB of persistent storage in each AWS Region at no additional cost. Persistent storage is located in your home directory (`$HOME`) and is private to you. Unlike ephemeral environment resources that are deleted after each shell session ends, data in your home directory persists between sessions. For more information about the retention of data in persistent storage, see [Persistent storage](#) in the *CloudShell User Guide*.

AWS Regions

In Lightsail, a CloudShell session will open in the AWS Region that provides the least latency to your physical location. This means that AWS Regions can change between sessions. Take note of which AWS Region--> your CloudShell session is located in so that you can use the 1 GB persistent storage. To change the session's AWS Region, choose the **Open in new browser tab** icon. This provides the option to access your CloudShell session in a new browser window.



In the navigation bar of the new browser tab, choose the name of the AWS Region that's currently displayed. Then choose the AWS Region that you want to switch to.



For more information about CloudShell, see the [CloudShell User Guide](#).

Launch and use AWS CloudShell

Learn how to launch and use an AWS CloudShell session within Lightsail. If you don't have permission to run CloudShell, you must add the `arn:aws:iam::aws:policy/`

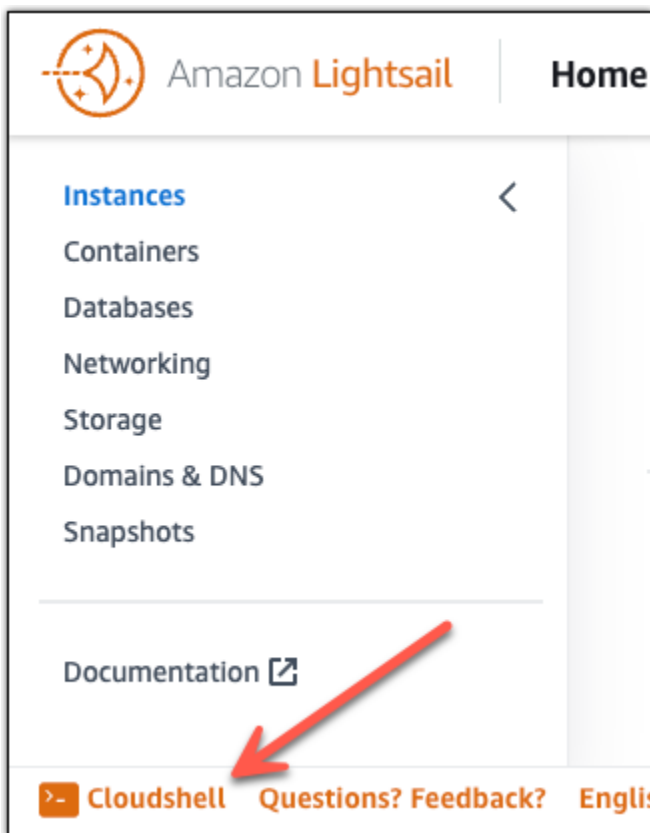
AWS`CloudShellFullAccess` policy to the AWS Identity and Access Management (IAM) identity that you're using. If you already have the `arn:aws:iam::aws:policy/AdministratorAccess` policy attached, you should be able to access CloudShell. For more information, see [???](#).

Launch AWS CloudShell

You can launch CloudShell from the Amazon Lightsail console. After the session begins, you can switch to your preferred shell, such as Bash, PowerShell, or Z shell.

Complete the following steps to launch a new AWS CloudShell session in Lightsail:

1. Sign in to the Lightsail console at <https://lightsail.aws.amazon.com/>.
2. Choose **CloudShell** on the Console Toolbar, in the lower left of the console. When the command prompt displays, the shell is ready for interaction.



3. (Optional) To choose a pre-installed shell to work with, enter one of the following program names at the command line prompt:

Bash: bash

If you switch to Bash, the symbol at the command prompt updates to \$. Bash is the default shell in AWS CloudShell.

PowerShell: pwsh

If you switch to PowerShell, the symbol at the command prompt updates to PS>.

Z shell: zsh

If you switch to Z shell, the symbol at the command prompt updates to %.

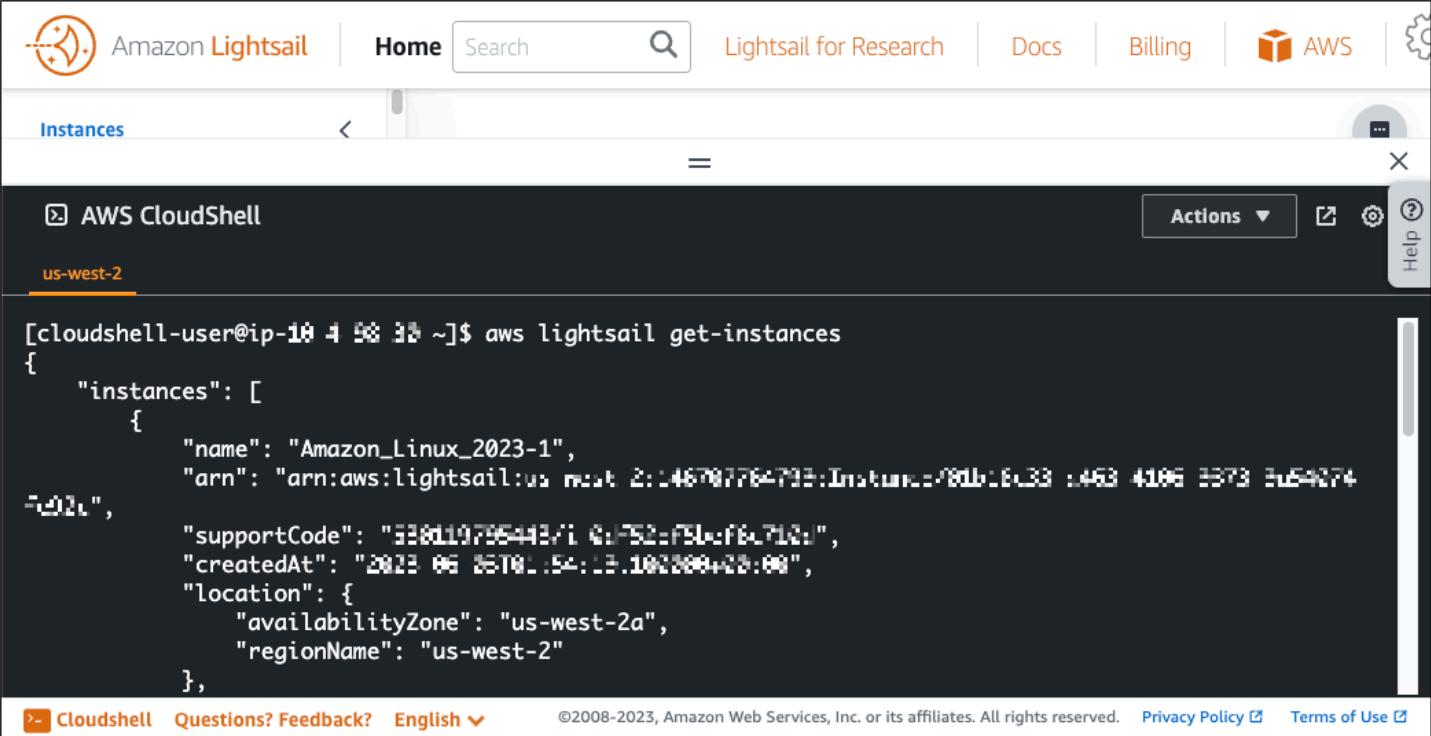
Example Example Lightsail API command in AWS CloudShell

There are multiple command line tools that are pre-installed on the CloudShell session for you to use. In this example, you use the Lightsail GetInstances API operation to view the instances that are in your Lightsail account. To learn more about the GetInstances API operation, see [GetInstances](#) in the *Amazon Lightsail API Reference*.

1. Sign in to the Lightsail console at <https://lightsail.aws.amazon.com/>.
2. Choose **CloudShell** on the Console Toolbar, in the lower left of the console.
3. Enter the following command after the AWS CloudShell prompt:

```
aws lightsail get-instances
```

You should now see a complete list of instances that are in your Lightsail account.



The screenshot shows the AWS CloudShell interface. At the top, there is a navigation bar with the Amazon Lightsail logo, a search bar, and links for Home, Lightsail for Research, Docs, Billing, and AWS. Below the navigation bar, the 'Instances' section is visible. The main area displays the AWS CloudShell terminal for the 'us-west-2' region. The terminal shows the command `aws lightsail get-instances` and its output, which is a JSON array of instance details. The output includes the instance name 'Amazon_Linux_2023-1', its ARN, support code, creation time, and location (availability zone 'us-west-2a' and region 'us-west-2').

```
[cloudshell-user@ip-10 4 58 38 ~]$ aws lightsail get-instances
{
  "instances": [
    {
      "name": "Amazon_Linux_2023-1",
      "arn": "arn:aws:lightsail:us-west-2:146707764795:Instance-f80b16c33-4453-4106-b373-2e54074",
      "supportCode": "338d1979644371-01752c791c-f6c712c",
      "createdAt": "2023-06-26T01:54:13.102000+00:00",
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      }
    }
  ],
}
```

Additional information

See the following documentation for more information about AWS CloudShell:

- [Amazon Lightsail API Reference](#)
- [Frequently asked questions in AWS CloudShell](#)
- [Supported browsers in AWS CloudShell](#)
- [Troubleshooting in AWS CloudShell](#)
- [Working with AWS services in AWS CloudShell](#)

Log Lightsail API calls with AWS CloudTrail

Amazon Lightsail is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Lightsail. CloudTrail captures all API calls for Lightsail as events. The calls captured include calls from the Lightsail console and code calls to the Lightsail API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Lightsail. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected

by CloudTrail, you can determine the request that was made to Lightsail, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

Lightsail Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Lightsail, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for Lightsail, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#)

All Lightsail actions are logged by CloudTrail and documented in the [Amazon Lightsail API Reference](#). For example, calls to the **GetInstance**, **AttachStaticIp** and **RebootInstance** sections generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentity Element](#).

Understanding Lightsail Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

Tutorial: Connect a Lightsail LAMP instance to an Aurora database

Application data for posts, pages, and users is stored on a MariaDB database that is running on your LAMP instance in Amazon Lightsail. If your instance fails, your data may become unrecoverable. To prevent this scenario, you should transfer your application data to a MySQL managed database.

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud. It combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open-source databases. Aurora is offered as part of the Amazon Relational Database Service (Amazon RDS). Amazon RDS is a managed database service that makes it easier to set up, operate, and scale a relational database in the cloud. For more information, see the [Amazon Relational Database Service User Guide](#) and the [Amazon Aurora User Guide for Aurora](#).

In this tutorial, we show you how to connect your application database from a LAMP instance in Lightsail to an Aurora managed database in Amazon RDS.

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Configure the security group for your Aurora database](#)
- [Step 3: Connect to your Aurora database from your Lightsail instance](#)
- [Step 4: Transfer the MariaDB database from your LAMP instance to your Aurora database](#)
- [Step 5: Configure your application to connect to your Aurora managed database](#)

Step 1: Complete the prerequisites

Complete the following prerequisites before you begin:

1. Create a LAMP instance in Lightsail, and configure your application on it. The instance should be in a running state before you continue. For more information, see [Tutorial: Launch and configure a LAMP instance in Lightsail](#).
2. Turn on VPC peering in your Lightsail account. For more information, see [Set up Amazon VPC peering to work with AWS resources outside of Lightsail](#).
3. Create an Aurora managed database in Amazon RDS. The database should be located in the same AWS Region as your LAMP instance. It should also be in a running state before you continue. For more information, see [Getting started with Amazon Aurora](#) in the *Amazon Aurora User Guide for Aurora*.

Step 2: Configure the security group for your Aurora database

An AWS security group acts as a virtual firewall for your AWS resources. It controls the incoming and outgoing traffic that can connect to your Aurora database in Amazon RDS. For more information about security groups, see [Control traffic to resources using security groups in the Amazon Virtual Private Cloud User Guide](#).

Complete the following procedure to configure the security group so that your LAMP instance can establish a connection to your Aurora database.

1. Sign in to the [Amazon RDS console](#).
2. Choose **Databases** in the navigation pane.
3. Choose the **Writer instance** of the Aurora database that your LAMP instance will connect to.
4. Choose the **Connectivity & security tab**.
5. In the **Endpoint & port** section, make a note of the **Endpoint name** and **Port** of the **Writer instance**. You will need these later when configuring your Lightsail instance to connect to the database.
6. In the **Security** section, choose the active VPC security group link. You will be redirected to your database's security group.

RDS > Databases > aurora-database-1 > aurora-database-1-instance-1

aurora-database-1-instance-1

Modify Actions

Related

Filter by databases

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU
aurora-database-1	Regional cluster	Aurora MySQL	us-west-2	1 instance	Available	-
aurora-database-1-instance-1	Writer instance	Aurora MySQL	us-west-2a	db.r5.large	Available	6.2

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance | Tags

Connectivity & security

Endpoint & port

Endpoint
aurora-database-1-instance-1.us-west-2.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-west-2a

VPC
vpc-...

Subnet group
default-vpc-...

Subnets
subnet-...
subnet-...
subnet-...

Security

VPC security groups
default (sg-...)
Active

Publicly accessible
Yes

Certificate authority
rds-ca-2019

Certificate authority date
August 22, 2024, 10:08 (UTC+10:08)

- Make sure that the security group for your Aurora database is selected.
- Choose the **Inbound rules** tab.
- Choose **Edit inbound rules**.

sg-... - default

Details | **Inbound rules** | Outbound rules | Tags

Inbound rules (3)

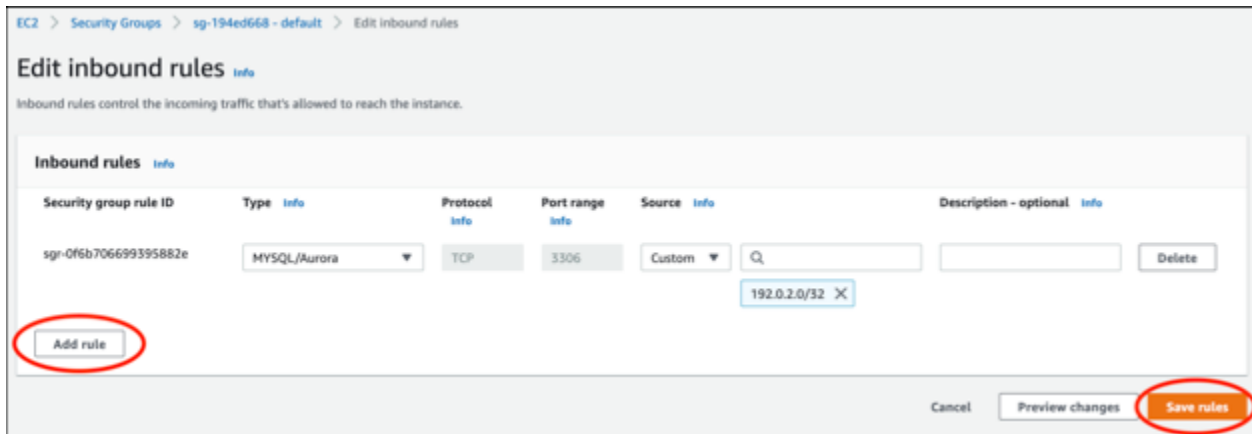
Filter security group rules

Manage tags Edit inbound rules

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-...	IPv4	SSH	TCP	22
-	sgr-...	IPv4	MYSQL/Aurora	TCP	3306
-	sgr-...	IPv6	SSH	TCP	22

- In the **Edit inbound rules** page, choose **Add rule**.
- Complete one of the following steps:

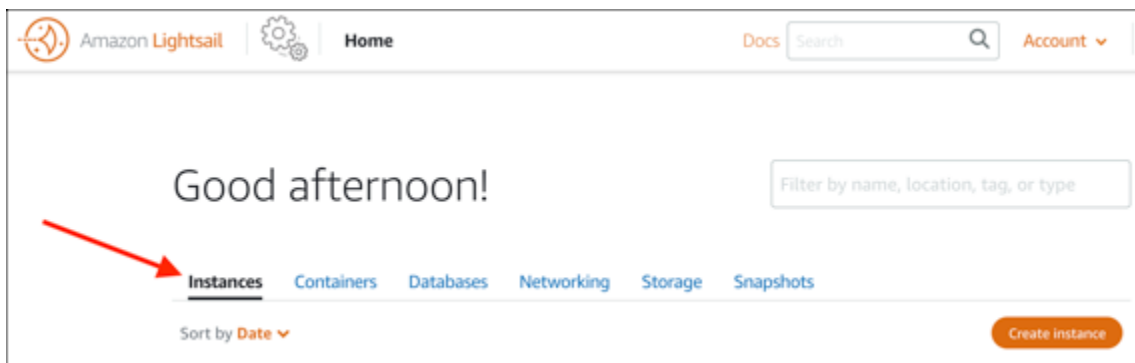
- If you are using the default MySQL port 3306, select **MySQL/Aurora** in the **Type** dropdown menu.
 - If you are using a custom port for your database, select **Custom TCP** in the **Type** dropdown menu and enter the port number in the **Port Range** text box.
12. In the **Source** text box, add the private IP address of your LAMP instance. You must enter the IP addresses in CIDR notation, which means that you must append /32. For example, to allow 192.0.2.0, enter 192.0.2.0/32.
 13. Choose **Save rules**.



Step 3: Connect to your Aurora database from your Lightsail instance

Complete the following procedure to confirm that you can connect to your Aurora database from your Lightsail instance.

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Instances** tab.



3. Choose the browser-based SSH client icon for your LAMP instance to connect to it using SSH.



4. After you're connected to your instance, enter the following command to connect to your Aurora database. In the command, replace *DatabaseEndpoint* with the endpoint address of your Aurora database, and replace *Port* with the port of your database. Replace *MyUserName* with the name of the user that you entered when creating the database.

```
mysql -h DatabaseEndpoint -P Port -u MyUserName -p
```

You should see a response similar to the following example, which confirms that your instance can access and connect to your Aurora database.

```
bitnami@ip-... $ mysql -h database.cluster-... .us-west-2.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 215
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> █
```

If you don't see this response, or you get an error message, then you might need to configure the security group of your database to allow the private IP address of your Lightsail instance to connect to it. For more information, see the [Configure the security group for your Aurora database](#) section of this guide.

Step 4: Transfer the MariaDB database from your LAMP instance to your Aurora database

Now that you've confirmed you can connect to your database from your instance, you should migrate the data from your LAMP instance database to your Aurora database. For more information, see [Migrating data to an Amazon Aurora MySQL DB cluster](#) in the *Amazon Aurora User Guide for Aurora*.

Step 5: Configure your application to connect to your Aurora managed database

After transferring your application data to your Aurora database, you should configure the application running on your LAMP instance to connect to your Aurora database. Connect to your LAMP instance using SSH, and access the application's database configuration file. In the configuration file, define the endpoint address of your Aurora database, the database user name, and password. Following is an example configuration file.

```
bitnami@ip-          :~/htdocs$ cat connectvalues.php
<?php
$host      = 'database.cluster-          .us-west-2.rds.amazonaws.com';
$username  = 'admin';
$password  = 'Password1';
```

Tutorial: How to create a HAR file

If you're experiencing difficulties with the Amazon Lightsail console or a Lightsail virtual private server (VPS), AWS Support might ask you to submit a HAR file from your web browser. A HAR file contains critical information that can help troubleshoot common, and hard to diagnose issues. The HAR file also allows AWS Support to investigate or replicate these issues.

Important

HAR files can capture sensitive information, such as user names, passwords, and keys. Be sure to remove any sensitive information from a HAR file before you share it.

In this guide, you will learn how to create a HAR file from your web browser. An HTTP Archive (HAR) file is a JSON file that contains the latest network activity recorded by your browser. Follow this step-by-step procedure to create a HAR file.

Contents

- [Step 1: Create a HAR file in your browser](#)
- [Step 2: Edit the HAR file to remove sensitive information](#)
- [Step 3: Submit the HAR file for review](#)

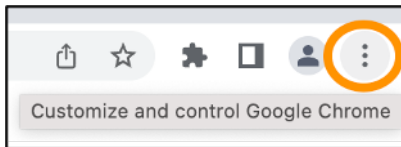
Step 1: Create a HAR file in your browser

Note

These instructions were last tested on Google Chrome version 101.0.4951.64, Microsoft Edge (Chromium) version 101.0.1210.47, and Mozilla Firefox version 91.9. Because these browsers are third-party products, these instructions might not match the experience in the latest versions or in the version that you use. In another browser, such as legacy Microsoft Edge (EdgeHTML) or Apple Safari for macOS, the process to generate a HAR file might be similar, but the steps will be different.

Google Chrome

1. In the browser, at the top right, choose **Customize and control Google Chrome**.

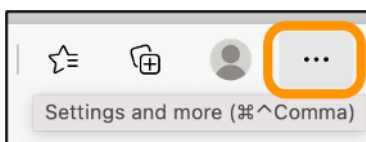


2. Pause on **More tools**, and then choose **Developer tools**.
3. With DevTools open in the browser, choose the **Network** panel.
4. Select the **Preserve log** check box.
5. Choose **Clear** to clear all current network requests.
6. Reproduce the issue you are facing
7. In DevTools, open the context (right-click) menu on any network request.
8. Choose **Save all as HAR with content**, and then save the file.

For more information, see [Open Chrome DevTools](#) and [Save all network requests to a HAR file](#) on the Google Developers website.

Microsoft Edge (Chromium)

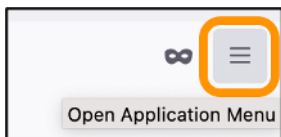
1. In the browser, at the top right, choose **Settings and more**.



2. Pause on **More tools**, and then choose **Developer tools**.
3. With DevTools open in the browser, choose the **Network** panel.
4. Select the **Preserve log** check box.
5. Choose **Clear** to clear all current network requests.
6. Reproduce the issue you are facing
7. In DevTools, open the context (right-click) menu on any network request.
8. Choose **Save all as HAR with content**, and then save the file.

Mozilla Firefox

1. In the browser, at the top right, choose **Open Application Menu**.



2. Choose **More tools**, and then choose **Web Developer tools**.
3. In the **Web Developer** menu, choose **Network**. (In some versions of Firefox, the **Web Developer** menu is in the **Tools** menu.)
4. Choose the gear icon, and then select **Persist Logs**.
5. Choose the trash can icon (**Clear**) to clear all current network requests.
6. Reproduce the issue you are facing.
7. In the Network Monitor, open the context menu (right-click) on any network request in the request list.
8. Choose **Save All As HAR**, and then save the file.

Step 2: Edit the HAR file to remove sensitive information

1. Open the HAR file in a text editor application.
2. Use the text editor's Find and Replace tools to identify and replace all sensitive information captured in the HAR file. This includes any user names, passwords, and keys that you entered in your browser while creating the file.
3. Save the edited HAR file with the sensitive information removed.

Step 3: Submit the HAR file for review

1. In the [AWS Support Center Console](#), under **Open support cases**, choose your support case.
2. In your support case, choose your preferred contact option, attach the edited HAR file, and then submit.

Force stop your Lightsail instance

Rarely, an instance can get stuck in the Stopping state. If this happens, there might be an issue with the underlying hardware that hosts your Lightsail instance. In this guide, you'll learn how to force stop an instance that's stuck in the stopping state. For more information about instance states, see [Start, Stop, or Restart your Amazon Lightsail instance](#).

How to force stop an instance

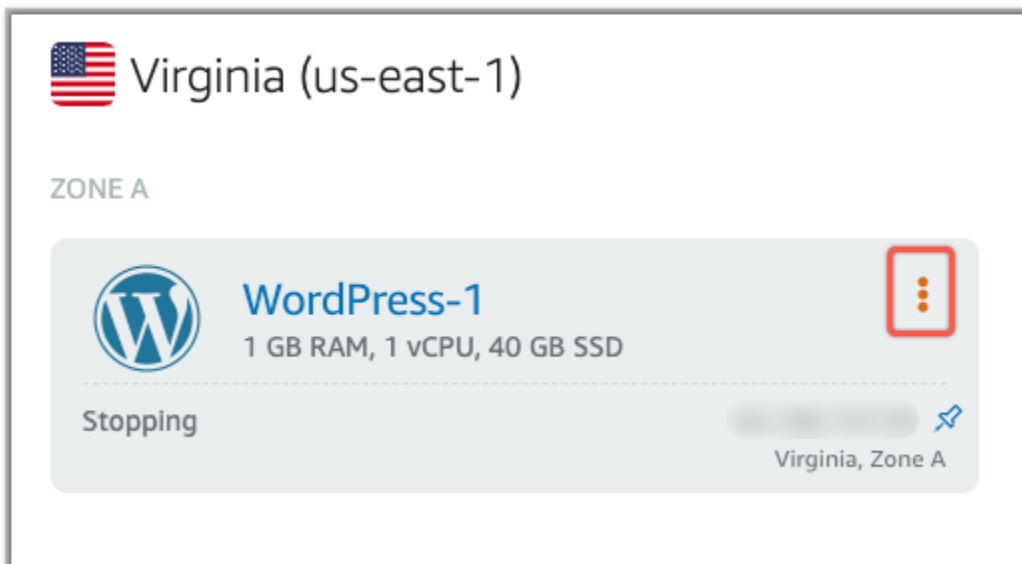
You can use the Lightsail console to force stop your instance, but only while the instance is in the stopping state. Alternatively, you can use the AWS Command Line Interface (AWS CLI) to force stop an instance while the instance is in any state except shutting-down and terminated. A force stop can take a few minutes to complete. If the instance hasn't stopped after 10 minutes, force stop it again.

When an instance is forced to stop, it doesn't have an opportunity to flush file system caches or file system metadata. After you force stop an instance, you should perform file system checks and repair procedures.

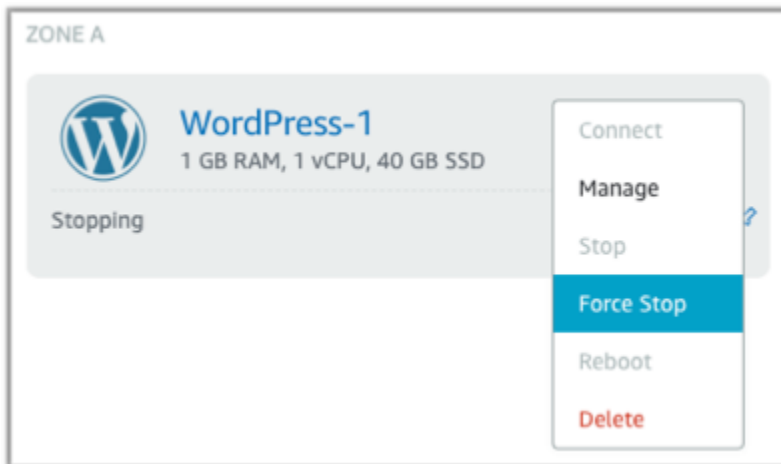
The following procedure explains the different ways that you can force stop a Lightsail instance.

Force stop an instance in the Lightsail console

1. Sign in to the [Lightsail console](#).
2. Choose the **Instances** tab.
3. Locate the instance that's stuck in the Stopping state. Then, choose the actions menu icon (⋮) displayed next to the instance name.



4. Choose **Force stop** in the dropdown list that appears.



Alternatively, you can choose the instance name to access the instance management page. Then, choose the **Force stop** button.



Force stop an instance with the AWS CLI

1. Before you begin, you need to install the AWS CLI. To learn more, see [Installing the AWS Command Line Interface](#). Be sure to [configure the AWS CLI](#) after you install it.
2. Use the [stop-instance](#) command and the `--force` parameter as follows:

```
aws lightsail stop-instance --instance-name WordPress-1 --force
```

Tutorial: Install Prometheus on a Linux-based Lightsail instance

Prometheus is an open source time series monitoring tool for managing a variety of system resources and applications. It provides a multidimensional data model, the ability to query the collected data, and detailed reporting and data visualization through Grafana.

By default, Prometheus is enabled to collect metrics on the server where it is installed. With the help of node exporters, metrics can be collected from other resources like web servers, containers, databases, custom applications, and other third-party systems. In this tutorial, we will show you how to install and configure Prometheus with node exporters on a Lightsail instance. For a full list of available exporters, see [Exporters and integrations](#) in the *Prometheus documentation*.

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Add users and local system directories to your Lightsail instance](#)
- [Step 3: Download the Prometheus binary packages](#)
- [Step 4: Configure Prometheus](#)
- [Step 5: Start Prometheus](#)
- [Step 6: Start Node Exporter](#)
- [Step 7: Configure Prometheus with the Node Exporter data collector](#)

Step 1: Complete the prerequisites

Before you can install Prometheus on an Amazon Lightsail instance, you must do the following:

- Create an instance in Lightsail. We recommend using the Ubuntu 20.04 LTS blueprint for your instance. For more information, see [Create an instance in Amazon Lightsail](#).

- Create and attach a static IP address to your new instance. For more information, see [Create a static IP address in Amazon Lightsail](#).
- Open ports 9090 and 9100 on the firewall of your new instance. Prometheus requires ports 9090 and 9100 to be open. For more information, see [Adding and editing instance firewall rules in Amazon Lightsail](#).

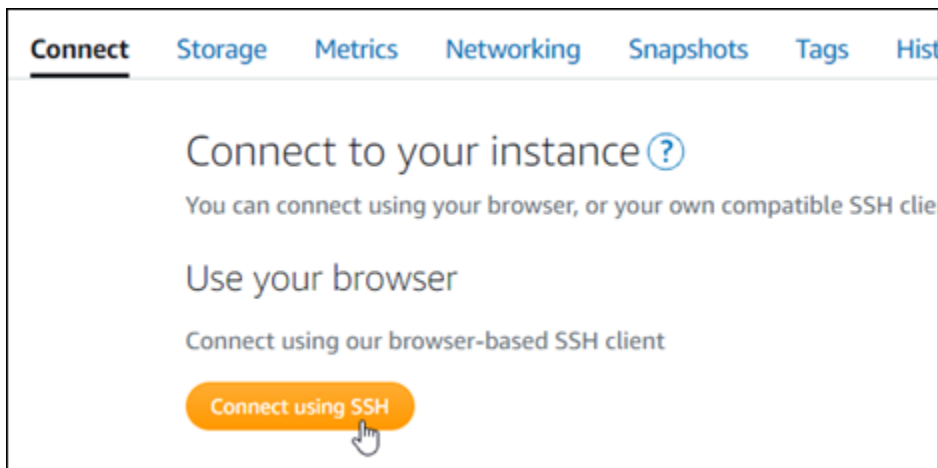
Step 2: Add users and local system directories to your Lightsail instance

Complete the following procedure to connect to your Lightsail instance using SSH and add users and system directories. This procedure creates the following Linux user accounts:

- `prometheus` – This account is used for installing and configuring the server environment.
- `exporter` – This account is used to configure the `node_exporter` extension.

These user accounts are created for the sole purpose of management and therefore do not require additional user services or permissions beyond the scope of this setup. In this procedure, you also create directories for storing and managing the files, service settings, and data that Prometheus uses to monitor resources.

1. Sign in to the [Lightsail console](#).
2. On your instance management page, under the **Connect** tab, choose **Connect using SSH**.



3. After you're connected, enter the following commands one by one to create two Linux user accounts, `prometheus` and `exporter`.

```
sudo useradd --no-create-home --shell /bin/false prometheus
```

```
sudo useradd --no-create-home --shell /bin/false exporter
```

4. Enter the following commands one by one to create local system directories.

```
sudo mkdir /etc/prometheus /var/lib/prometheus
```

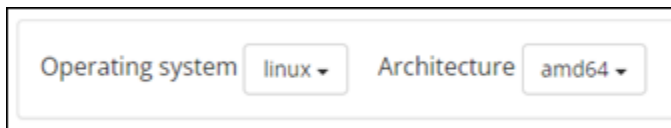
```
sudo chown prometheus:prometheus /etc/prometheus
```

```
sudo chown prometheus:prometheus /var/lib/prometheus
```

Step 3: Download the Prometheus binary packages

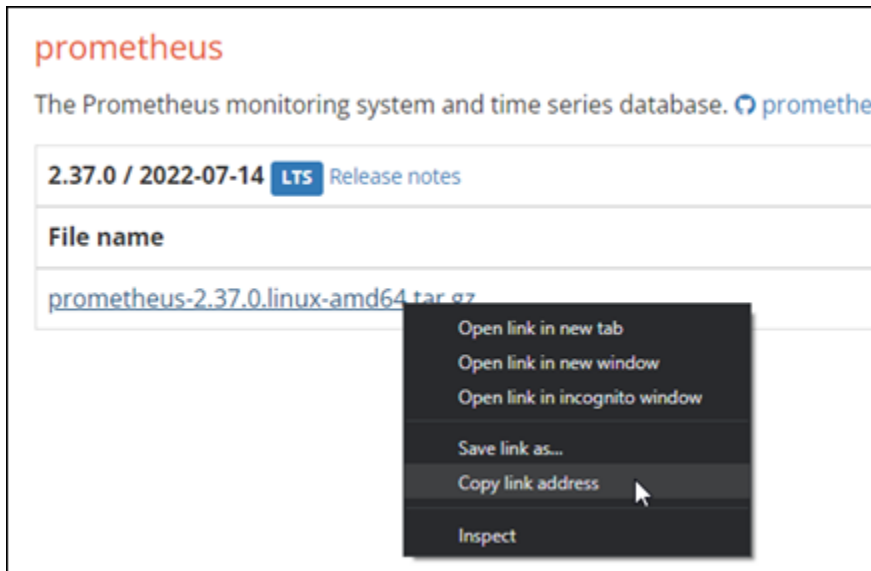
Complete the following procedure to download the Prometheus binary packages to your Lightsail instance.

1. Open a web browser on your local computer and browse to the [Prometheus downloads page](#).
2. At the top of the page, for the **Operating system** dropdown, select **linux**. For **Architecture**, select **amd64**.



The image shows a screenshot of the Prometheus download page. At the top, there are two dropdown menus. The first is labeled 'Operating system' and has 'linux' selected. The second is labeled 'Architecture' and has 'amd64' selected. The entire selection area is enclosed in a thin black border.

3. Choose or right-click the **Prometheus** download link that appears, and copy the link address to a text file on your computer. Do the same for the **node_exporter** download link that appears. You will use both copied addresses later in this procedure.



4. Connect to your Lightsail instance using SSH.
5. Enter the following command to change directories to your home directory.

```
cd ~
```

6. Enter the following command to download the Prometheus binary packages to your instance.

```
curl -LO prometheus-download-address
```

Replace *prometheus-download-address* with the address that you copied earlier in this procedure. The command should look like the following example when you add the address.

```
curl -LO https://github.com/prometheus/prometheus/releases/download/v2.37.0/prometheus-2.37.0.linux-amd64.tar.gz
```

7. Enter the following command to download the `node_exporter` binary packages to your instance.

```
curl -LO node_exporter-download-address
```

Replace *node_exporter-download-address* with the address that you copied in the previous step of this procedure. The command should look like the following example when you add the address.

```
curl -LO https://github.com/prometheus/node\_exporter/releases/download/v1.3.1/node\_exporter-1.3.1.linux-amd64.tar.gz
```

8. Run the following commands one by one to extract the contents of the downloaded Prometheus and Node Exporter files.

```
tar -xvf prometheus-2.37.0.linux-amd64.tar.gz
```

```
tar -xvf node_exporter-1.3.1.linux-amd64.tar.gz
```

Several subdirectories are created after the contents of the downloaded files are extracted.

9. Enter the following commands one by one to copy the `prometheus` and `promtool` extracted files to the `/usr/local/bin` programs directory.

```
sudo cp -p ./prometheus-2.37.0.linux-amd64/prometheus /usr/local/bin
```

```
sudo cp -p ./prometheus-2.37.0.linux-amd64/promtool /usr/local/bin
```

10. Enter the following command to change the ownership of the `prometheus` and `promtool` files to the `prometheus` user that you created earlier in this tutorial.

```
sudo chown prometheus:prometheus /usr/local/bin/prom*
```

11. Enter the following commands one by one to copy the `consoles` and `console_libraries` subdirectories to `/etc/prometheus`. The `-r` option performs a recursive copy of all directories within the hierarchy.

```
sudo cp -r ./prometheus-2.37.0.linux-amd64/consoles /etc/prometheus
```

```
sudo cp -r ./prometheus-2.37.0.linux-amd64/console_libraries /etc/prometheus
```

12. Enter the following commands one by one to change the ownership of the copied files to the `prometheus` user that you created earlier in this tutorial. The `-R` option performs a recursive ownership change for all of the files and directories within the hierarchy.

```
sudo chown -R prometheus:prometheus /etc/prometheus/consoles
```

```
sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries
```

13. Enter the following commands one by one to copy the configuration file `prometheus.yml` to the `/etc/prometheus` directory and change the ownership of the copied file to the `prometheus` user that you created earlier in this tutorial.

```
sudo cp -p ./prometheus-2.37.0.linux-amd64/prometheus.yml /etc/prometheus
```

```
sudo chown prometheus:prometheus /etc/prometheus/prometheus.yml
```

14. Enter the following command to copy the `node_exporter` file from the `./node_exporter*` subdirectory to the `/usr/local/bin` programs directory.

```
sudo cp -p ./node_exporter-1.3.1.linux-amd64/node_exporter /usr/local/bin
```

15. Enter the following command to change the ownership of the file to the `exporter` user that you created earlier in this tutorial.

```
sudo chown exporter:exporter /usr/local/bin/node_exporter
```

Step 4: Configure Prometheus

Complete the following procedure to configure Prometheus. In this procedure, you open and edit the `prometheus.yml` file, which contains various settings for the Prometheus tool. Prometheus establishes a monitoring environment based on the settings that you configure in the file.

1. Connect to your Lightsail instance using SSH.
2. Enter the following command to create a backup copy of the `prometheus.yml` file before you open and edit it.

```
sudo cp /etc/prometheus/prometheus.yml /etc/prometheus/prometheus.yml.backup
```

3. Enter the following command to open the `prometheus.yml` file using Vim.

```
sudo vim /etc/prometheus/prometheus.yml
```

Following are a few important parameters that you might want to configure in the `prometheus.yml` file:

- `scrape_interval` — Located under the `global` header, this parameter defines the time interval (in seconds) for how often Prometheus will collect or *scrape* metric data for a given target. As indicated by the `global` tag, this setting is universal for all resources that Prometheus monitors. This setting also applies for exporters, unless an individual exporter provides a different value that overrides the global value. You can keep this parameter set to its current value of 15 seconds.
- `job_name` — Located under the `scrape_configs` header, this parameter is a label that identifies exporters in the result set of a data query or visual display. You can specify the value of a job name to best reflect the resources that are being monitored in your environment. For example, you can label a job for managing a website as `business-web-app`, or you can label a database as `mysql-db-1`. In this initial setup, you are only monitoring the Prometheus server, so you can keep the current `prometheus` value.
- `targets` — Located under the `static_configs` header, the `targets` setting uses an `ip_addr:port` key-value pair to identify the location where a given exporter is running. You will change the default setting in steps 4–7 of this procedure.

```
my global config
global:
  A scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  B # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

  C static_configs:
    - targets: ["localhost:9090"]
```


Note

For this initial setup, you don't need to configure the `alerting` and `rule_files` parameters.

- In the `prometheus.yml` file that you have open in Vim, press the **I** key to enter insert mode in Vim.
- Scroll and find the `targets` parameter located under the `static_configs` header.
- Change the default setting to `<ip_addr>:9090`. Replace `<ip_addr>` with the static IP address of the instance. The modified parameter should look like the following example.

```
static_configs:
- targets: ["192.0.2.0:9090"]
```

- Press the **Esc** key to exit insert mode, and type `:wq!` to save your changes and quit Vim.
- (Optional) If something went wrong, enter the following command to replace the `prometheus.yml` file with the backup that you created earlier in this procedure.

```
sudo cp /etc/prometheus/prometheus.yml.backup /etc/prometheus/prometheus.yml
```

Step 5: Start Prometheus

Complete the following procedure to start the Prometheus service on your instance.

- Connect to your Lightsail instance using SSH.
- Enter the following command to start the Prometheus service.

```
sudo -u prometheus /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus --web.console.templates=/etc/prometheus/conssoles --web.console.libraries=/etc/prometheus/console_libraries
```

The command line outputs details on the startup process and other services. It should also indicate that the service is listening on port 9090.

```
ts=2022-06-02T15:46:09.3362 caller=main.go:993 level=info fs_type=EXT4_SUPER_MAGIC
ts=2022-06-02T15:46:09.3362 caller=main.go:996 level=info msg="TSDB started"
ts=2022-06-02T15:46:09.3362 caller=main.go:1177 level=info msg="Loading configuration file" filename=/etc/prometheus/prometheus.yml
ts=2022-06-02T15:46:09.3452 caller=main.go:1214 level=info msg="Completed loading of configuration file" filename=/etc/prometheus/prometheus.yml
totalDuration=8.392805ms db_storage=1.681µs remote_storage=2.294µs web_handler=1.213µs query_engine=1.435µs scrape=7.967101ms scrape_sd=48.64µs notify=1.931µs notify_sd=2.455µs rules=2.669µs tracing=6.382µs
ts=2022-06-02T15:46:09.3452 caller=main.go:957 level=info msg="Server is ready to receive web requests."
ts=2022-06-02T15:46:09.3452 caller=manager.go:937 level=info component="rule manager" msg="Starting rule manager..."
```

If the service doesn't start, see the [Step 1: Complete the prerequisites](#) section of this tutorial for information about creating instance firewall rules to allow traffic on this port. For other errors, review the `prometheus.yml` file to confirm that there are no syntax errors.

3. After the running service is validated, press **Ctrl+C** to stop it.
4. Enter the following command to open the `systemd` configuration file in Vim. This file is used to start Prometheus.

```
sudo vim /etc/systemd/system/prometheus.service
```

5. Insert the following lines into the file.

```
[Unit]
Description=PromServer
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
--config.file /etc/prometheus/prometheus.yml \
--storage.tsdb.path /var/lib/prometheus/ \
--web.console.templates=/etc/prometheus/consoles \
--web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

The preceding instructions are used by the Linux `systemd` service manager to start Prometheus on the server. When invoked, Prometheus runs as the `prometheus` user and references the `prometheus.yml` file for loading the configuration settings and storing the time series data in the `/var/lib/prometheus` directory. You can run `man systemd` from the command line to see more information about the service.

6. Press the **Esc** key to exit insert mode, and type **:wq!** to save your changes and quit Vim.
7. Enter the following command to load the information into the `systemd` service manager.

```
sudo systemctl daemon-reload
```

8. Enter the following command to restart Prometheus.

```
sudo systemctl start prometheus
```

9. Enter the following command to check the status of the Prometheus service.

```
sudo systemctl status prometheus
```

If the service launched properly, you receive an output similar to the following example.

```
ubuntu@ip-172-26-11-178:~$ sudo systemctl status prometheus
• prometheus.service - PrometheusServer
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-06-02 16:03:33 UTC; 2s ago
     Main PID: 105938 (prometheus)
        Tasks: 6 (Limit: 1164)
       Memory: 39.3M
      CGroup: /system.slice/prometheus.service
             └─105938 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/
```

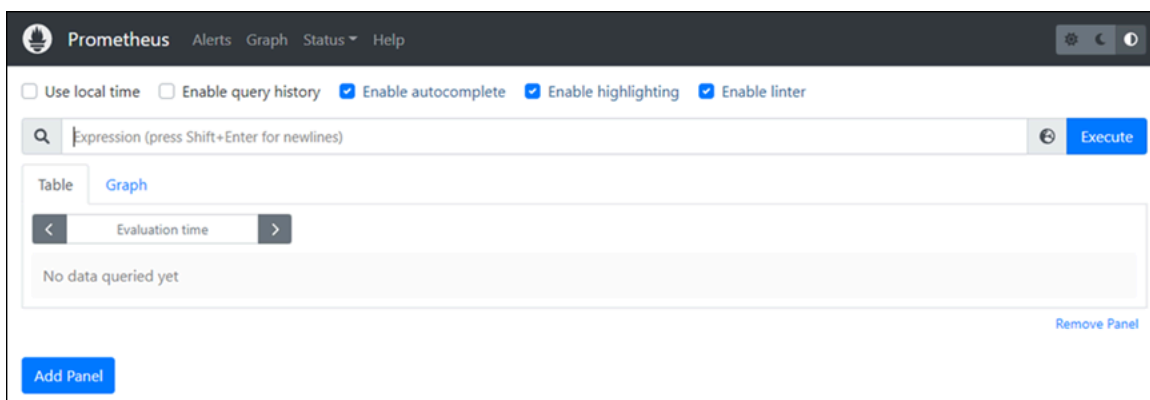
10. Press **Q** to exit the status command.
11. Enter the following command to enable Prometheus to start when the instance is booted.

```
sudo systemctl enable prometheus
```

12. Open a web browser on your local computer and go to the following web address to view the Prometheus management interface.

```
http:<ip_addr>:9090
```

Replace *<ip_addr>* with the static IP address of your Lightsail instance. You should see a dashboard similar to the following example.



Step 6: Start Node Exporter

Complete the following procedure to start the Node Exporter service.

1. Connect to your Lightsail instance using SSH.
2. Enter the following command to create a systemd service file for `node_exporter` using Vim.

```
sudo vim /etc/systemd/system/node_exporter.service
```

3. Press the **I** key to enter insert mode in Vim.
4. Add the following lines of text into the file. This will configure `node_exporter` with monitoring collectors for CPU load, file system usage, and memory resources.

```
[Unit]
Description=NodeExporter
Wants=network-online.target
After=network-online.target

[Service]
User=exporter
Group=exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter --collector.disable-defaults \
--collector.meminfo \
--collector.loadavg \
--collector.filesystem

[Install]
WantedBy=multi-user.target
```

Note

These instructions disable default machine metrics for Node Exporter. For a complete list of metrics available for Ubuntu, see the [Prometheus node_exporter man page](#) in the *Ubuntu documentation*.

5. Press the **Esc** key to exit insert mode, and type **:wq!** to save your changes and quit Vim.
6. Enter the following command to reload the systemd process.

```
sudo systemctl daemon-reload
```

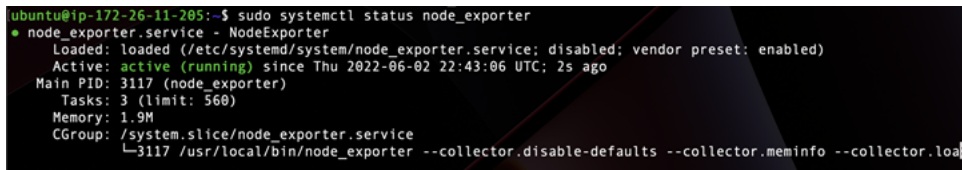
7. Enter the following command to start the `node_exporter` service.

```
sudo systemctl start node_exporter
```

8. Enter the following command to check the status of the `node_exporter` service.

```
sudo systemctl status node_exporter
```

If the service launched successfully, you receive an output similar to the following example.



```
ubuntu@ip-172-26-11-205:~$ sudo systemctl status node_exporter
● node_exporter.service - NodeExporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; disabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-06-02 22:43:06 UTC; 2s ago
     Main PID: 3117 (node_exporter)
        Tasks: 3 (limit: 500)
       Memory: 1.9M
      CGroup: /system.slice/node_exporter.service
             └─3117 /usr/local/bin/node_exporter --collector.disable-defaults --collector.meminfo --collector.loa
```

9. Press **Q** to exit the status command.
10. Enter the following command to enable Node Exporter to start when the instance is booted.

```
sudo systemctl enable node_exporter
```

Step 7: Configure Prometheus with the Node Exporter data collector

Complete the following procedure to configure Prometheus with the Node Exporter data collector. You do this by adding a new `job_name` parameter for `node_exporter` in the `prometheus.yml` file.

1. Connect to your Lightsail instance using SSH.
2. Enter the following command to open the `prometheus.yml` file using Vim.

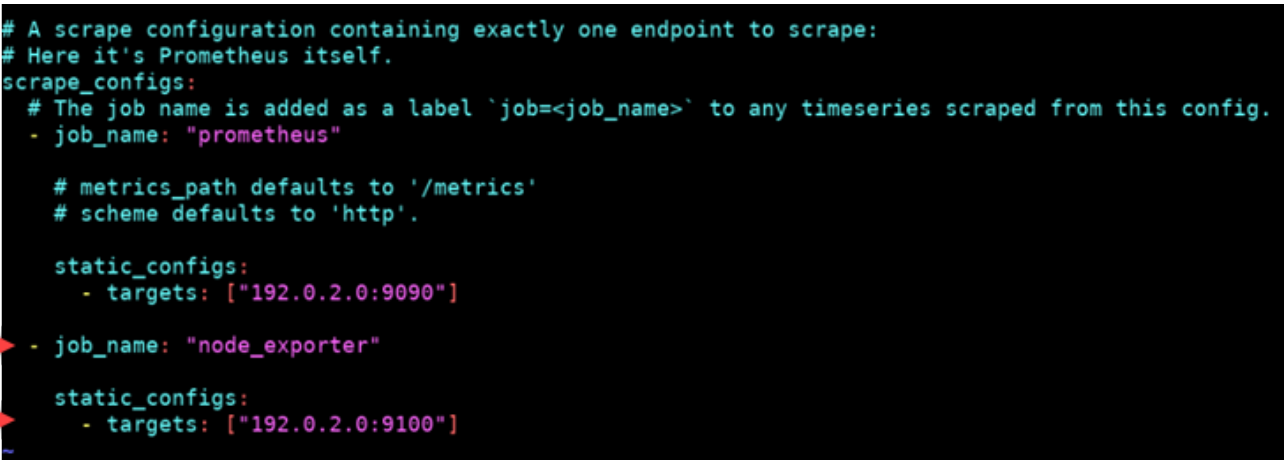
```
sudo vim /etc/prometheus/prometheus.yml
```

3. Press the **I** key to enter insert mode in Vim.
4. Add the following lines of text into the file, below the existing `- targets:` `["<ip_addr>:9090"]` parameter.

```
- job_name: "node_exporter"
```

```
static_configs:  
- targets: ["<ip_addr>:9100"]
```

The modified parameter in the `prometheus.yml` file should look like the following example.



```
# A scrape configuration containing exactly one endpoint to scrape:  
# Here it's Prometheus itself.  
scrape_configs:  
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.  
  - job_name: "prometheus"  
  
    # metrics_path defaults to '/metrics'  
    # scheme defaults to 'http'.  
  
    static_configs:  
      - targets: ["192.0.2.0:9090"]  
  
  - job_name: "node_exporter"  
  
    static_configs:  
      - targets: ["192.0.2.0:9100"]
```

Note the following:

- Node Exporter listens to port 9100 for the Prometheus server to scrape the data. Confirm that you followed the steps for creating instance firewall rules as outlined in the [Step 1: Complete the prerequisites](#) section of this tutorial.
 - As with the configuration of the Prometheus `job_name`, replace `<ip_addr>` with the static IP address that's attached to your Lightsail instance.
5. Press the **Esc** key to exit insert mode, and type `:wq!` to save your changes and quit Vim.
 6. Enter the following command to restart the Prometheus service so that the changes to the configuration file can take effect.

```
sudo systemctl restart prometheus
```

7. Enter the following command to check the status of the Prometheus service.

```
sudo systemctl status prometheus
```

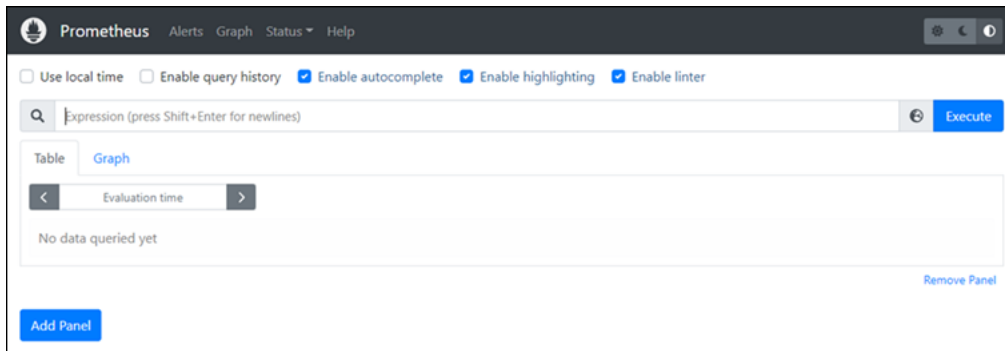
If the service restarted properly, you receive output similar to the following.

```
ubuntu@ip-172-26-11-170:~$ sudo systemctl status prometheus
● prometheus.service - PrometheusServer
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-06-02 16:03:33 UTC; 2s ago
     Main PID: 105938 (prometheus)
        Tasks: 6 (Limit: 1164)
       Memory: 39.3M
      CGroup: /system.slice/prometheus.service
              └─105938 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/
```

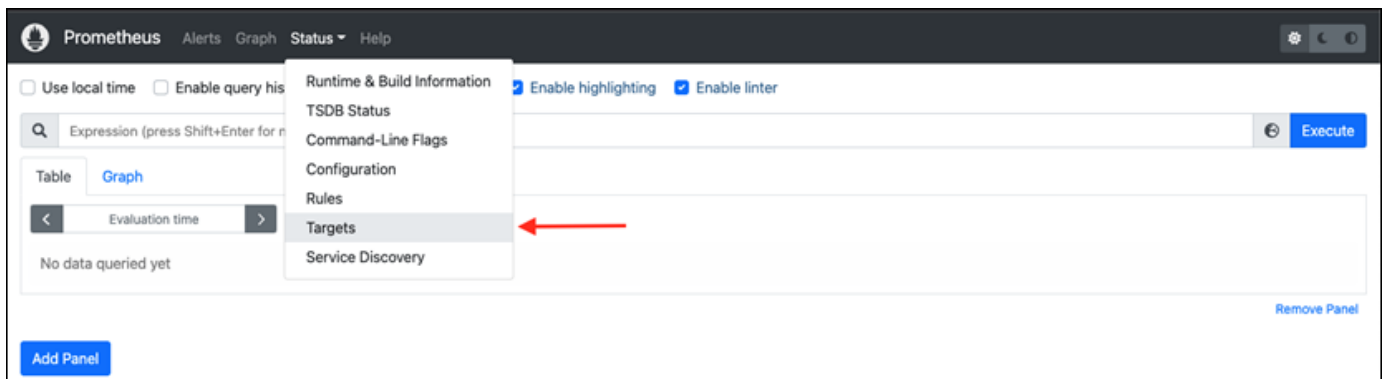
8. Press **Q** to exit the status command.
9. Open a web browser on your local computer and go to the following web address to view the Prometheus management interface.

`http:<ip_addr>:9090`

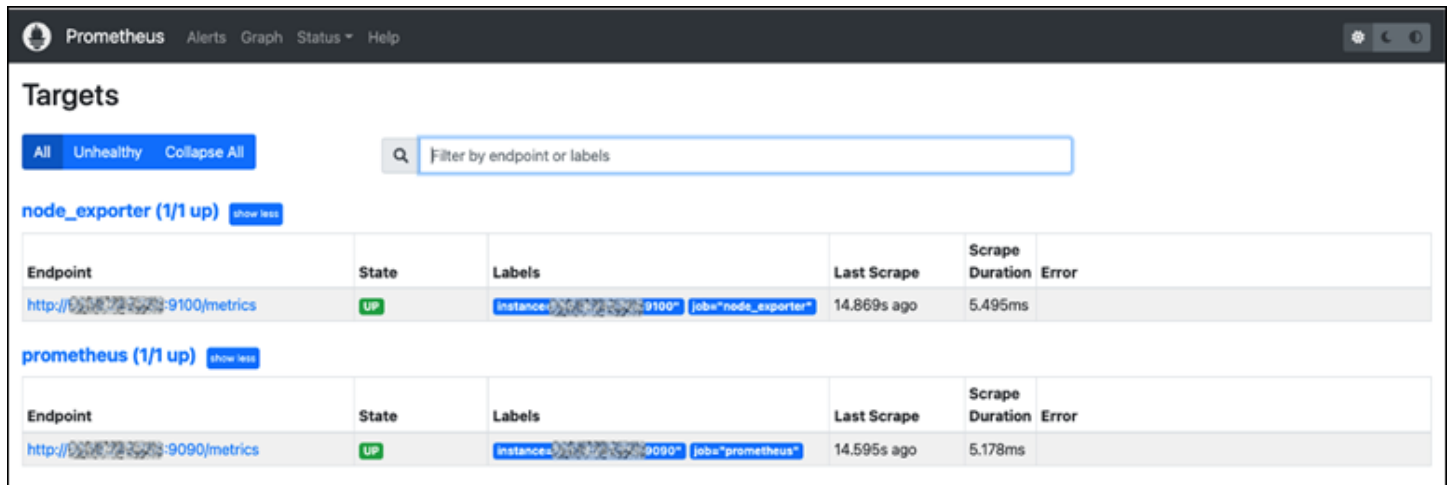
Replace *<ip_addr>* with the static IP address of your Lightsail instance. You should see a dashboard similar to the following example.



10. In the main menu, choose the **Status** dropdown and select **Targets**.



On the next screen, you should see two targets. The first target is for the **node_exporter** metrics collector job, and the second target is for the **prometheus** job.



The screenshot shows the Prometheus web interface. At the top, there's a navigation bar with 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below that, the 'Targets' section is visible. It includes a search bar labeled 'Filter by endpoint or labels' and two target groups. The first group is 'node_exporter (1/1 up)' with a 'show less' button. It contains one target with the following details:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://[redacted]:9100/metrics	UP	instance="[redacted]:9100" job="node_exporter"	14.869s ago	5.495ms	

The second group is 'prometheus (1/1 up)' with a 'show less' button. It contains one target with the following details:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://[redacted]:9090/metrics	UP	instance="[redacted]:9090" job="prometheus"	14.595s ago	5.178ms	

The environment is now properly set up for collecting metrics and monitoring the server.

Tutorial: Launch and configure a Lightsail LAMP instance

Amazon Lightsail is the easiest way to get started with Amazon Web Services (AWS) if you just need virtual private servers. Lightsail includes everything you need to launch your project quickly – a virtual machine, SSD-based storage, data transfer, DNS management, and a static IP – for a low, predictable price.

This tutorial shows you how to launch and configure a LAMP instance on Lightsail. It includes steps to connect to your instance via SSH, get the application password for your instance, create a static IP and attach it to your instance, and create a DNS zone and map your domain. When you're done with this tutorial, you have the fundamentals to get your instance up and running on Lightsail.

Contents

- [Step 1: Sign up for AWS](#)
- [Step 2: Create a LAMP instance](#)
- [Step 3: Connect to your instance via SSH and get the application password for your LAMP instance](#)
- [Step 4: Install an application on top of your LAMP instance](#)
- [Step 5: Create a static IP address and attach it to your LAMP instance](#)
- [Step 6: Create a DNS zone and map a domain to your LAMP instance](#)
- [Next steps](#)

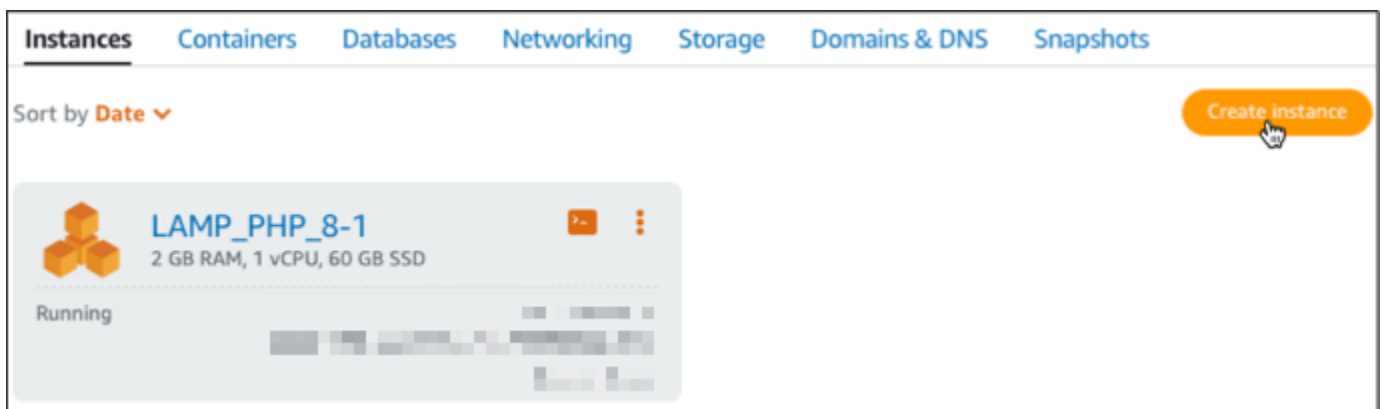
Step 1: Sign up for AWS

This tutorial requires an AWS account. [Sign up for AWS](#), or [sign in to AWS](#) if you already have an account.

Step 2: Create a LAMP instance

Get your LAMP instance up and running in Lightsail. For more information about creating an instance in Lightsail, see [Creating an Amazon Lightsail instance in the Lightsail](#) documentation.

1. Sign in to the [Lightsail console](#).
2. On the **Instances** tab of the Lightsail home page, choose **Create instance**.

















3. Choose the AWS Region and Availability Zone for your instance.





Select your instance location

Select a Region

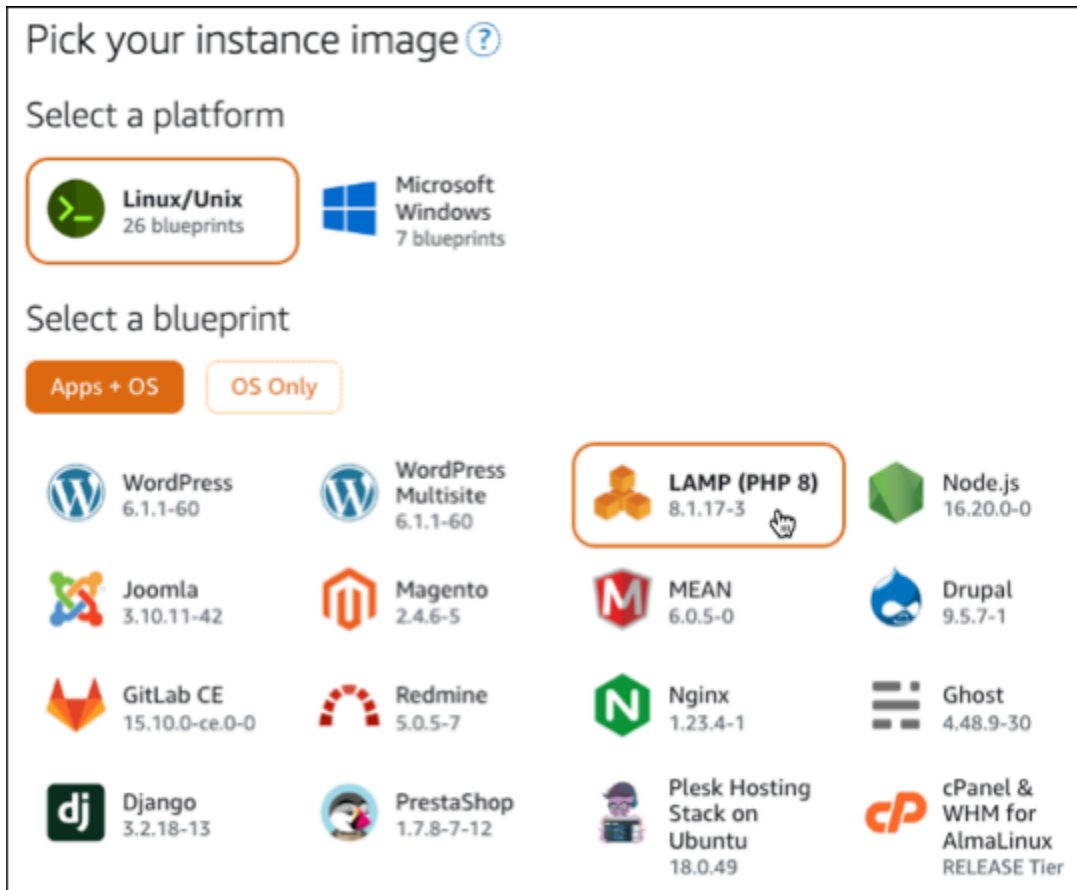
The closer your instance is to your users, the less latency they will experience.
[Learn more about Regions](#)

 Oregon us-west-2	 Ohio us-east-2	 Virginia us-east-1	 Montreal ca-central-1
 Tokyo ap-northeast-1	 Seoul ap-northeast-2	 Ireland eu-west-1	 Sydney ap-southeast-2
 London eu-west-2	 Paris eu-west-3	 Frankfurt eu-central-1	 Singapore ap-southeast-1
 Mumbai ap-south-1	 Stockholm eu-north-1		

Select an Availability Zone

 Zone A us-west-2a	 Zone B us-west-2b	 Zone C us-west-2c	 Zone D us-west-2d
---	---	---	---

4. Choose your instance image.
 - a. Choose **Linux/Unix** as the platform.
 - b. Choose **LAMP (PHP 8)** as the blueprint.



5. Choose an instance plan.

A plan includes a low, predictable cost, machine configuration (RAM, SSD, vCPU), and data transfer allowance. You can try the \$3.50 USD Lightsail plan without charge for one month (up to 750 hours). AWS credits one free month to your account.

Note

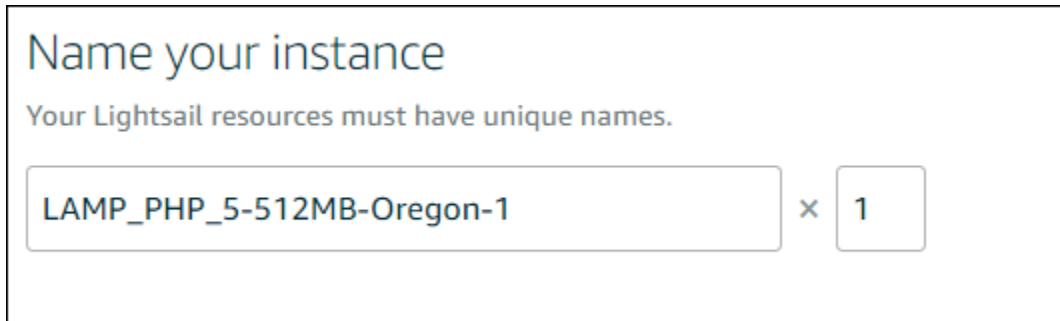
As part of the AWS Free Tier, you can get started with Amazon Lightsail for free on select instance bundles. For more information, see **AWS Free Tier** on the [Amazon Lightsail Pricing page](#).

6. Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.

- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.



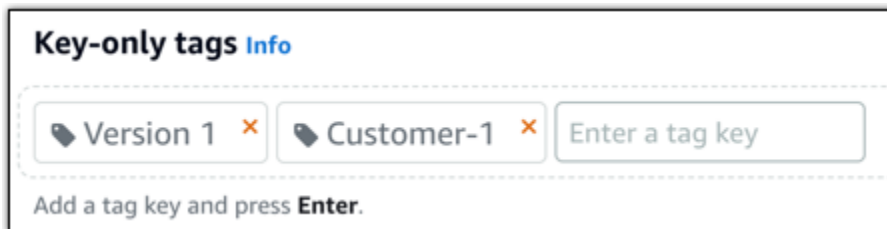
Name your instance

Your Lightsail resources must have unique names.

LAMP_PHP_5-512MB-Oregon-1 × 1

7. Choose one of the following options to add tags to your instance:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



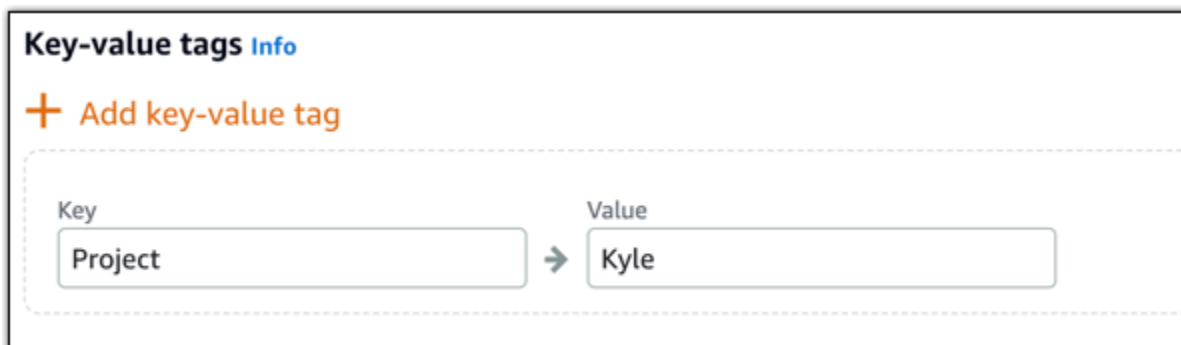
Key-only tags Info

Version 1 × Customer-1 × Enter a tag key

Add a tag key and press **Enter**.

- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key Value

Project → Kyle

Note

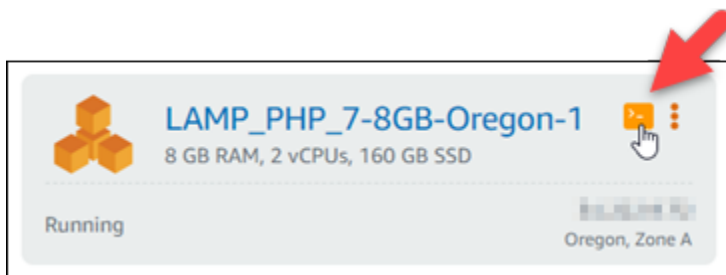
For more information about key-only and key-value tags, see [Tags](#).

8. Choose **Create instance**.

Step 3: Connect to your instance via SSH and get the application password for your LAMP instance

The default password to sign in to your database in LAMP is stored on your instance. Retrieve it by connecting to your instance using the browser-based SSH terminal in the Lightsail console and running a special command. For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

1. On the **Instances** tab of the Lightsail home page, choose the SSH quick-connect icon for your LAMP instance.



2. After the browser-based SSH client window opens, enter the following command to retrieve the default application password:

```
cat bitnami_application_password
```

Note

If you're in a directory other than the user home directory, then enter `cat $HOME/bitnami_application_password`.

3. Make note of the password displayed on the screen. You use this password later to install Bitnami applications on your instance, or to access the MySQL database with the user name of `root`.

```
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1065-aws x86_64)
*** System restart required ***

  _____
 | _   _  |
 | | | | |
 | |_| | |
 |  _  |
 | |_| | |
 |  _  |
 |_____| |

*** Welcome to the Bitnami LAMP 5.6.37-2 ***
*** Documentation: https://docs.bitnami.com/aws/infrastructure/lamp/ ***
***                 https://docs.bitnami.com/aws/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bitnami@ip-10-0-1-1:~$ cat bitnami_application_password
pSAqtrn2l9nt
bitnami@ip-10-0-1-1:~$
```

Step 4: Install an application on top of your LAMP instance

Deploy your PHP application on top of your LAMP instance, or install a Bitnami application. The main directory to deploy your PHP application is `/opt/bitnami/apache2/htdocs`. Copy your PHP application files to that directory and access the application by browsing to your instance's public IP address.

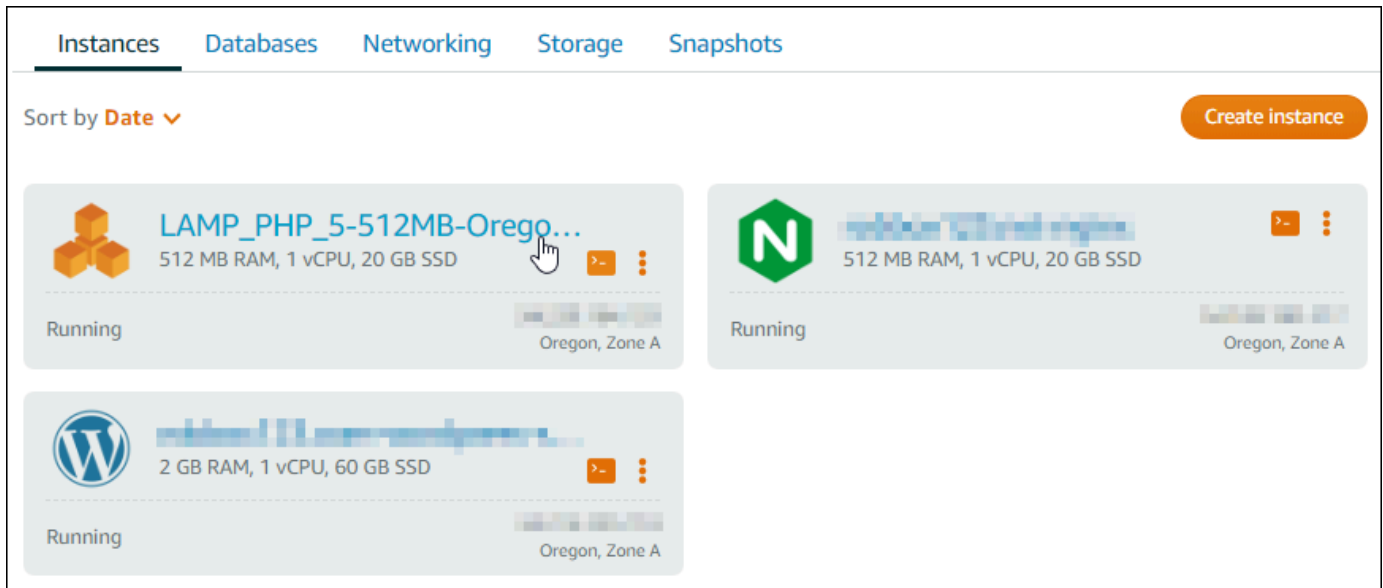
You can also install a Bitnami application using module installers. Download WordPress, Drupal, Magento, Moodle among other applications from the [Bitnami website](https://docs.bitnami.com/aws/infrastructure/lamp/) and extend the functionality of your server. For more information about installing Bitnami applications, see [Getting Started](#) in the Bitnami documentation.

Step 5: Create a static IP address and attach it to your LAMP instance

The default public IP for your LAMP instance changes if you stop and start the instance. A static IP address, attached to an instance, stays the same even if you stop and start your instance.

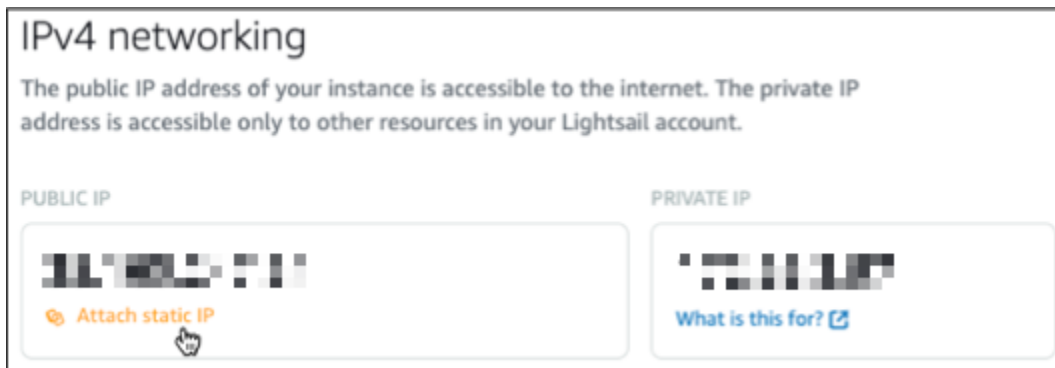
Create a static IP address and attach it to your LAMP instance. For more information, see [Create a static IP and attach it to an instance](#) in the Lightsail documentation.

1. On the **Instances** tab of the Lightsail home page, choose your running LAMP instance.



The screenshot shows the Amazon Lightsail console with the 'Instances' tab selected. At the top, there are navigation tabs for 'Instances', 'Databases', 'Networking', 'Storage', and 'Snapshots'. Below the tabs, there is a 'Sort by Date' dropdown and a 'Create instance' button. Three instance cards are visible, each showing a status of 'Running' and the location 'Oregon, Zone A'. The first instance is named 'LAMP_PHP_5-512MB-Orego...' with 512 MB RAM, 1 vCPU, and 20 GB SSD. The second instance is named 'WordPress 5.3.2-512MB-Orego...' with 512 MB RAM, 1 vCPU, and 20 GB SSD. The third instance is named 'WordPress 5.3.2-2GB-Orego...' with 2 GB RAM, 1 vCPU, and 60 GB SSD.

2. Choose the **Networking** tab, then choose **Attach static IP**.



The screenshot shows the 'IPv4 networking' settings page. It explains that the public IP address is accessible to the internet, while the private IP address is accessible only to other resources in the account. There are two sections: 'PUBLIC IP' and 'PRIVATE IP'. The 'PUBLIC IP' section shows a blurred IP address and an 'Attach static IP' button. The 'PRIVATE IP' section shows a blurred IP address and a 'What is this for?' link.

3. Name your static IP, then choose **Create and attach**.



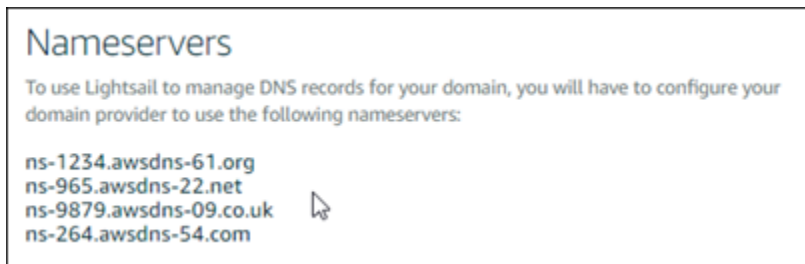
The screenshot shows the 'Create and attach a static IP' dialog box. It instructs the user to create and attach a static IP as a stable endpoint before assigning a domain to 'LAMP_PHP_8-1'. The 'Identify your static IP' section has a text input field containing 'StaticIp-1'. Below the input field, there is a note: 'Name can contain letters and numbers; hyphen (-), period (.) and underscore (_) characters can separate words.' At the bottom, there are two buttons: 'Cancel' and 'Create and attach'.

Step 6: Create a DNS zone and map a domain to your LAMP instance

Transfer management of your domain's DNS records to Lightsail. This allows you to more easily map a domain to your LAMP instance, and manage all of your website's resources using the Lightsail console. For more information, see [Creating a DNS zone to manage your domain's DNS records](#).

1. On the **Domains & DNS** tab of the Lightsail home page, choose **Create DNS zone**.
2. Enter your domain, then choose **Create DNS zone**.
3. Make note of the name server addresses listed on the page.

You add these name server addresses to your domain name's registrar to transfer management of your domain's DNS records to Lightsail.



4. After management of your domain's DNS records are transferred to Lightsail, add an A record to point the apex of your domain to your LAMP instance, as follows:
 - a. In the **Assignments** tab of the DNS zone, choose **Add assignment**.
 - b. In the **Select a domain** field, choose the domain or subdomain.
 - c. In the **Select a resource** drop down, select the LAMP instance you created earlier in this tutorial.
 - d. Choose the **Assign**.

Allow time for the change to propagate through the internet's DNS before your domain begins routing traffic to your LAMP instance.

Next steps

Here are a few additional steps you can perform after launching a LAMP instance in Amazon Lightsail:

- [Create a snapshot of your Linux or Unix instance](#)

- [Create and attach additional block storage disks to your Linux-based instances](#)

Tutorial: Launch and configure a Windows Server 2016 instance

Amazon Lightsail is the easiest way to get started with Amazon Web Services (AWS) if you just need virtual private servers. Lightsail includes everything you need to launch your project quickly – a virtual machine, SSD-based storage, data transfer, DNS management, and a static IP – for a low, predictable price.

This tutorial shows you how to launch and configure a Windows Server 2016 instance on Lightsail. It includes steps to connect to your instance via RDP, create a static IP and attach it to your instance, and create a DNS zone and map your domain. When you're done with this tutorial, you have the fundamentals to get your instance up and running on Lightsail.

Contents

- [Step 1: Sign up for AWS](#)
- [Step 2: Create a Windows Server 2016 instance](#)
- [Step 3: Connect to your Windows Server 2016 instance with RDP](#)
- [Step 4: Create a static IP address and attach it to your Windows Server 2016 instance](#)
- [Step 5: Create a DNS zone and map a domain to your Windows Server 2016 instance](#)
- [Next steps](#)

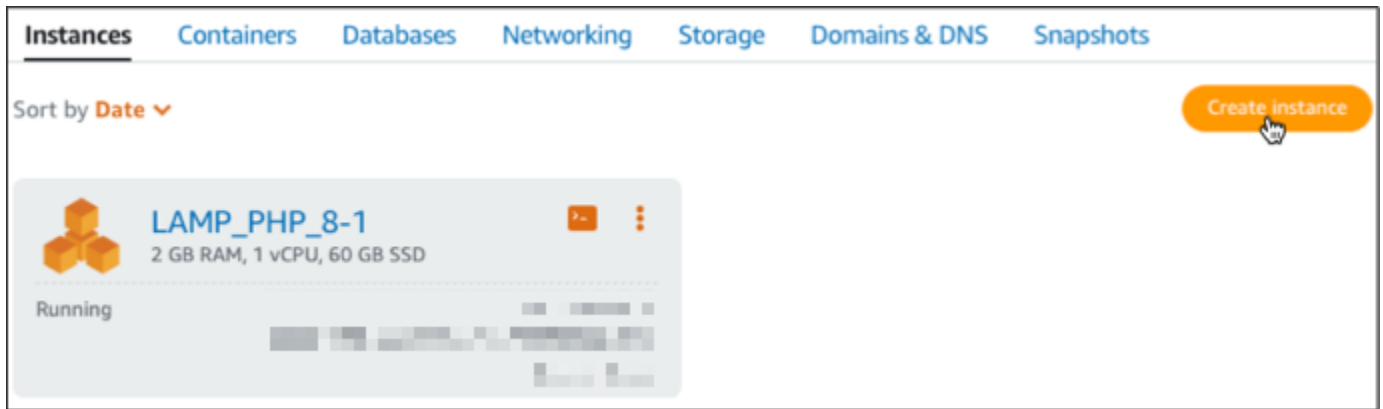
Step 1: Sign up for AWS

This tutorial requires an AWS account. [Sign up for AWS](#), or [sign in to AWS](#) if you already have an account.

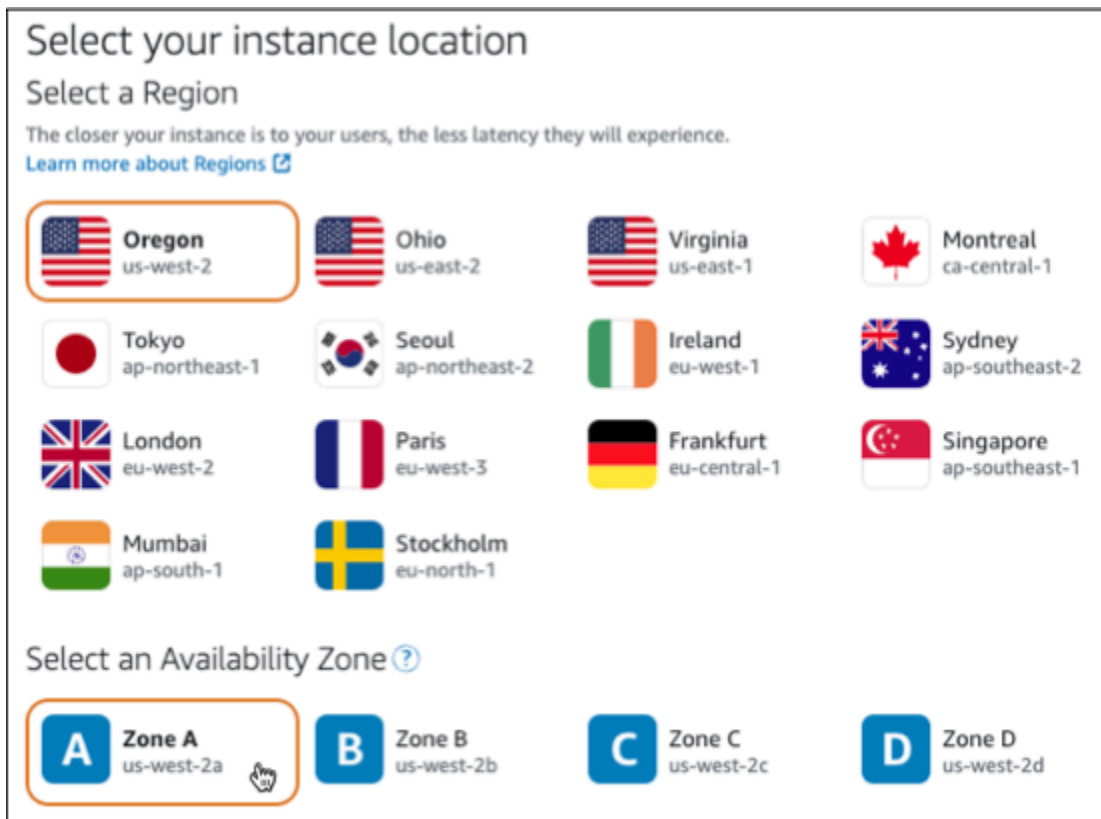
Step 2: Create a Windows Server 2016 instance in Lightsail

Get your Windows Server 2016 instance up and running in Lightsail. For more information, see [Get started with Windows Server-based instances](#).

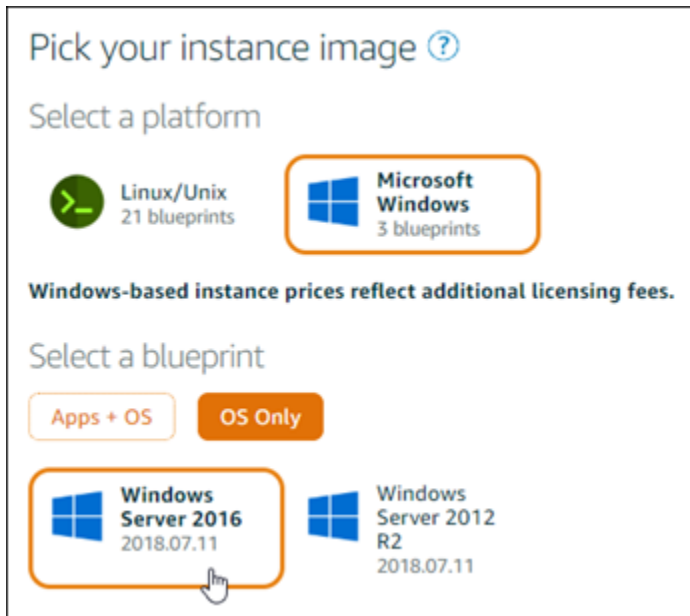
1. Sign in to the [Lightsail console](#).
2. On the **Instances** tab of the Lightsail home page, choose **Create instance**.



3. Choose the AWS Region and Availability Zone for your instance.



4. Choose your instance image.
 - a. Choose **Microsoft Windows** as the platform.
 - b. Choose **OS Only**, then choose **Windows Server 2016** as the blueprint.



5. Choose an instance plan.

A plan includes a low, predictable cost, machine configuration (RAM, SSD, vCPU), and data transfer allowance. You can try the \$8 USD Lightsail plan without charge for one month (up to 750 hours). AWS credits one free month to your account.

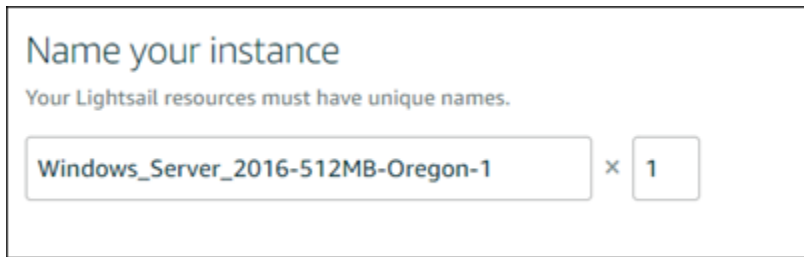
Note

As part of the AWS Free Tier, you can get started with Amazon Lightsail for free on select instance bundles. For more information, see **AWS Free Tier** on the [Amazon Lightsail Pricing page](#).

6. Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.



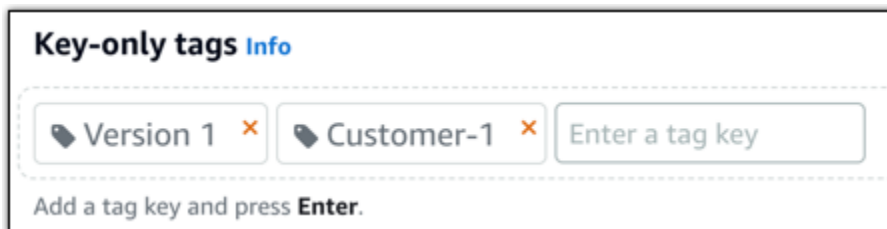
Name your instance

Your Lightsail resources must have unique names.

Windows_Server_2016-512MB-Oregon-1 x 1

7. Choose one of the following options to add tags to your instance:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



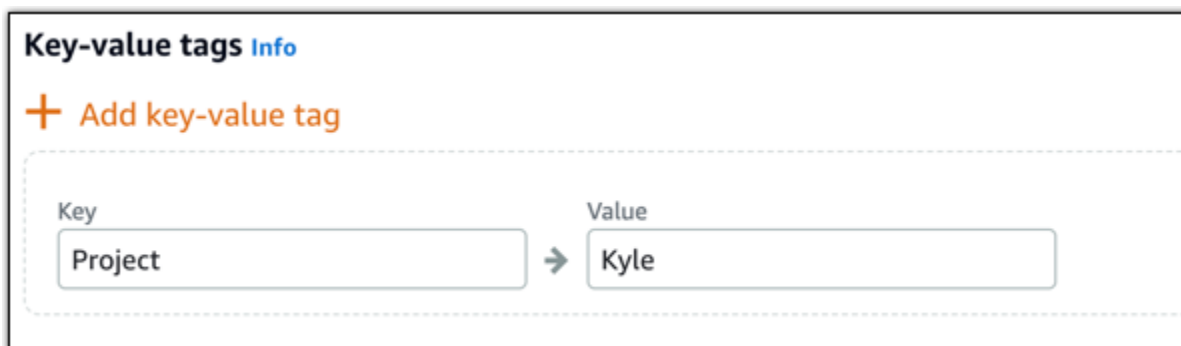
Key-only tags Info

Version 1 x Customer-1 x Enter a tag key

Add a tag key and press **Enter**.

- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags Info

+ Add key-value tag

Key Value

Project → Kyle

Add a tag key and press **Enter**.

Note

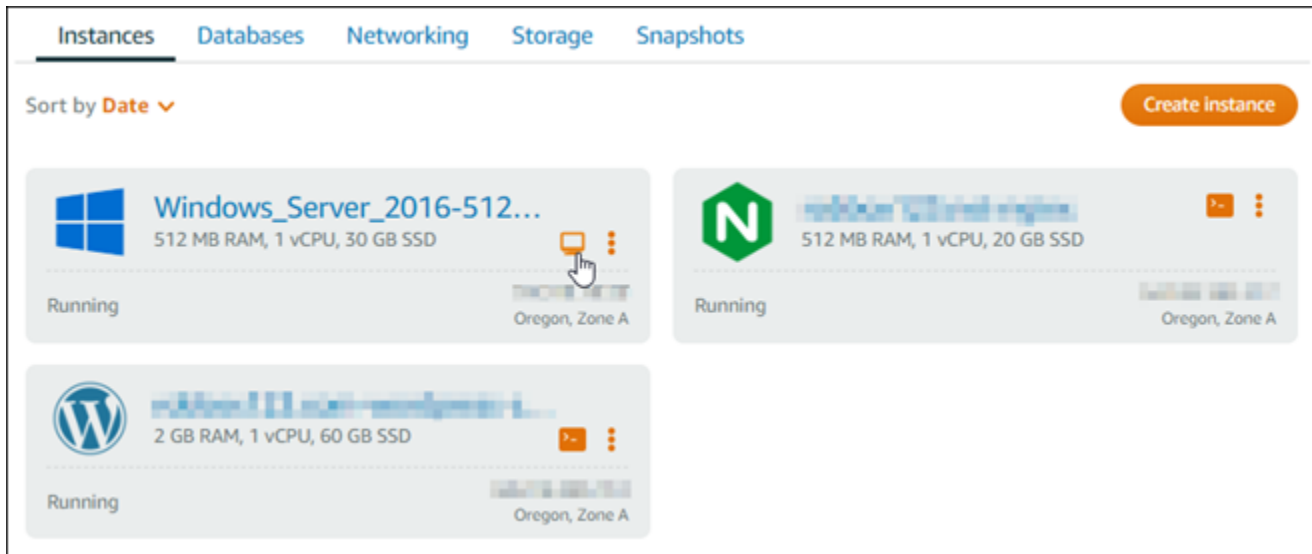
For more information about key-only and key-value tags, see [Tags](#).

8. Choose **Create instance**.

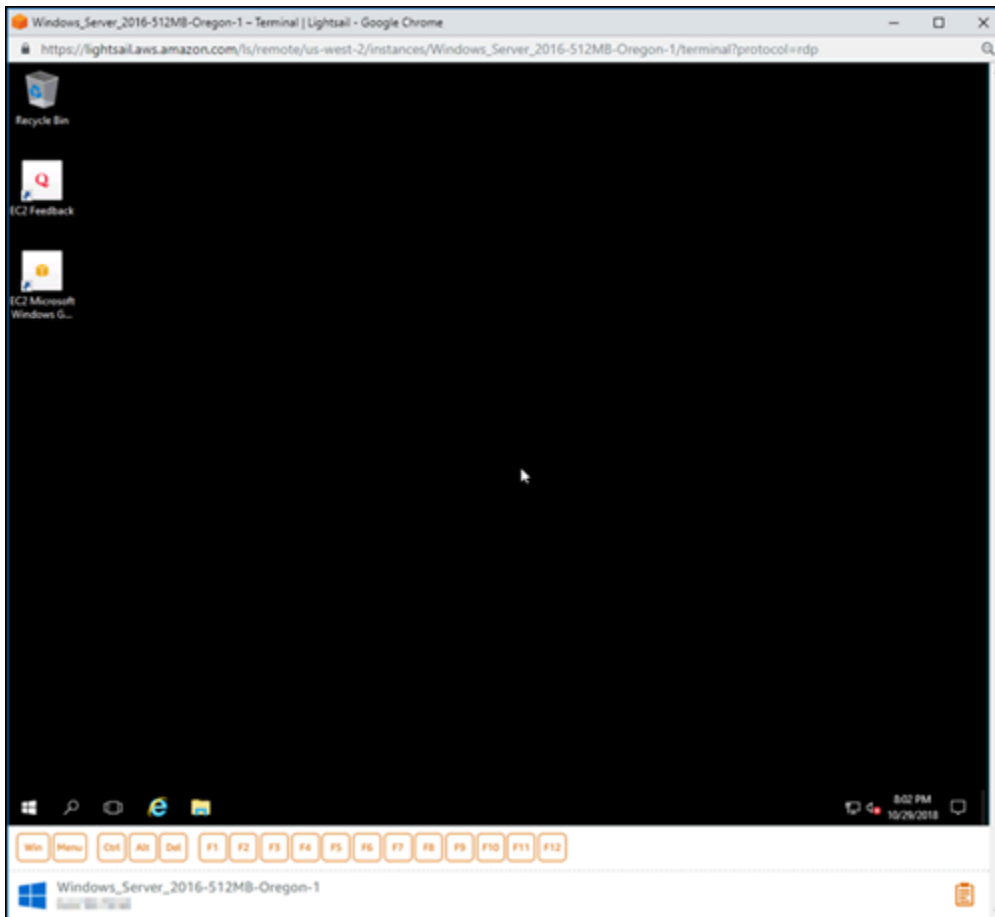
Step 3: Connect to your Windows Server 2016 instance with RDP

Connect to your Windows Server 2016 instance using the browser-based RDP client in the Lightsail console. For more information, see [Connect to your Windows instance](#).

1. On the **Instances** tab of the Lightsail home page, choose the RDP quick-connect icon for your Windows Server 2016 instance.



2. After the browser-based RDP client window opens, you can begin configuring your Windows Server 2016 instance:

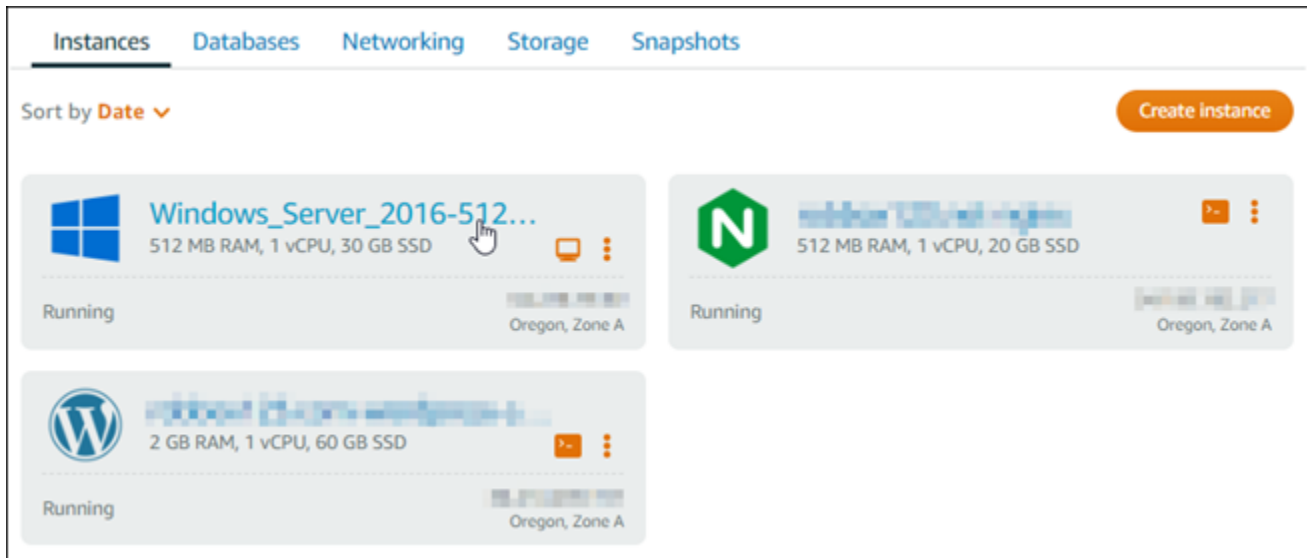


Step 4: Create a static IP address and attach it to your Windows Server 2016 instance

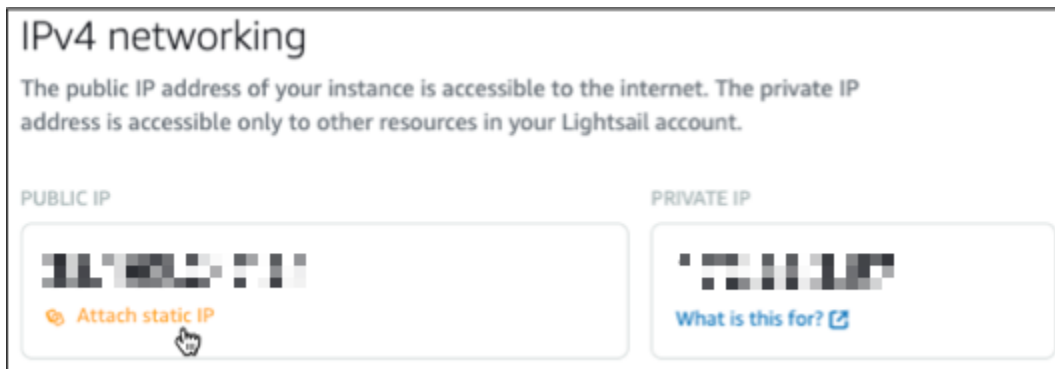
The default public IP for your Windows Server 2016 instance changes if you stop and start the instance. A static IP address, attached to an instance, stays the same even if you stop and start your instance.

Create a static IP address and attach it to your Windows Server 2016 instance. For more information, see [Create a static IP and attach it to an instance](#) in the Lightsail documentation.

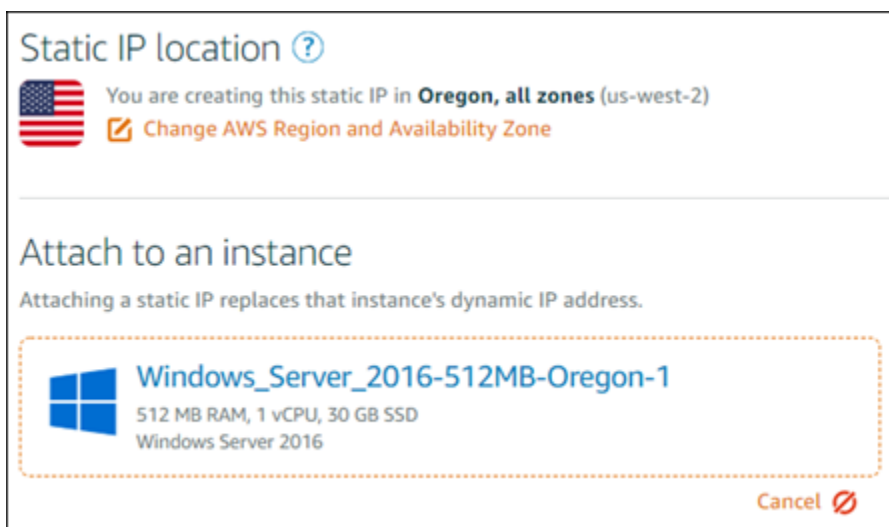
1. On the **Instances** tab of the Lightsail home page, choose your running Windows Server 2016 instance.



2. Choose the **Networking** tab, then choose **Create static IP**.



3. The static IP location, and attached instance are pre-selected based on the instance that you chose earlier in this tutorial.



4. Enter a name for your static IP.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

5. Choose **Create**.

Create and attach a static IP

Create and attach a **Static IP** as a stable endpoint before assigning a domain to **LAMP_PHP_8-1**.

Identify your static IP

Your Lightsail resources must have unique names.

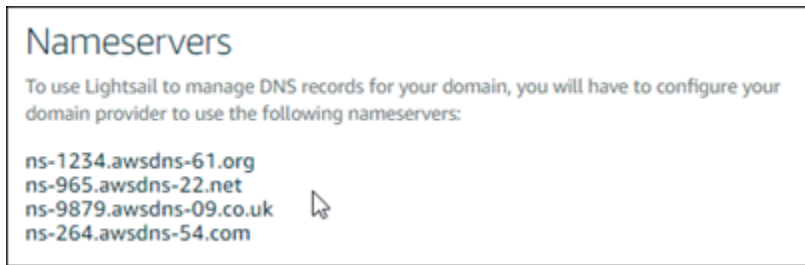
Name can contain letters and numbers; hyphen (-), period (.) and underscore (_) characters can separate words.

Step 5: Create a DNS zone and map a domain to your Windows Server 2016 instance

Transfer management of your domain's DNS records to Lightsail. This allows you to more easily map a domain to your Windows Server 2016 instance, and manage all of your website's resources using the Lightsail console. For more information, see [Create a DNS zone to manage your domain's DNS records](#) in the Lightsail documentation.

1. On the **Domains & DNS** tab of the Lightsail home page, choose **Create DNS zone**.
2. Enter your domain, then choose **Create DNS zone**.
3. Make note of the name server addresses listed on the page.

You add these name server addresses to your domain name's registrar to transfer management of your domain's DNS records to Lightsail.



4. After management of your domain's DNS records are transferred to Lightsail, add an A record to point the apex of your domain to your LAMP instance, as follows:
 - a. In the **Assignments** tab of the DNS zone, choose **Add assignment**.
 - b. In the **Select a domain** field, choose the domain or subdomain.
 - c. In the **Select a resource** drop down, select the LAMP instance you created earlier in this tutorial.
 - d. Choose the **Assign**.

Allow time for the change to propagate through the internet's DNS before your domain begins routing traffic to your LAMP instance.

Next steps

Here are a few additional steps you can perform after launching a Windows Server 2016 instance in Amazon Lightsail:

- [Creating a snapshot of your Windows Server instance](#)
- [Best practices for securing Windows Server-based Lightsail instances](#)
- [Creating and attaching a block storage disk to your Windows Server instance](#)
- [Extending the storage space of your Windows Server instance](#)

Learn more about Amazon Lightsail

The following list includes links to additional information for Amazon Lightsail that is not published in the Lightsail User Guide.

Contents

- [Blogs](#)

- [Tutorials](#)
- [Videos](#)

Blogs

- [Monitoring the health of Amazon Lightsail instances with Datadog](#)

March 30, 2022 – Explore how monitoring Lightsail workloads with Datadog can help you ensure application performance and control costs.

- [How to set up Galaxy for research on AWS using Amazon Lightsail](#)

January 13, 2022 – Deploy Galaxy, a scientific workflow, data integration, and digital preservation platform on Lightsail.

- [What happens when you type a URL into your browser](#)

August 26, 2021 – What happens when you type a URL into your browser and press enter?

- [Monitoring memory usage in Amazon Lightsail instance](#)

June 14, 2021 – Configure a Lightsail instance to send memory usage to Amazon CloudWatch for monitoring, alarming, and notifications.

- [Frictionless hosting of containerized ASP.NET web apps using Amazon Lightsail](#)

June 10, 2021 – How to take a containerized ASP.NET web application that connects to a PostgreSQL database and deploy it to Lightsail.

- [Launching a WordPress website using Amazon Lightsail containers](#)

April 5, 2021 – Launch a WordPress website using Lightsail containers and a Lightsail database.

- [Lightsail containers: an easy way to run your containers in the cloud](#)

November 13, 2020 – Deploy your container-based workloads on Lightsail.

- [Migrating web services from Amazon Lightsail to Amazon EC2](#)

October 16, 2020 – Set up a production environment in Amazon EC2 and migrate a web service into that environment from Lightsail.

- [Building a Graylog server to run on an Amazon Lightsail instance](#)

July 28, 2020 – How to build a Graylog server on Lightsail.

- [Improving website performance with Lightsail content delivery network](#)

July 23, 2020 – Configure Lightsail distribution to work with both a standard web server in addition to WordPress.

- [Proactively monitoring system performance on Amazon Lightsail instances](#)

June 4, 2020 – Configure a burstable capacity alert so you can prevent system performance issues before they impact your users.

- [Enhancing site security with new Lightsail firewall features](#)

May 7, 2020 – Restrict remote access with SSH to a single source IP address.

- [Using CodeDeploy and CodePipeline to deploy applications to Amazon Lightsail](#)

April 23, 2020 – Configure Lightsail to work with CodeDeploy and CodePipeline to automatically deploy (or update) an application every time you push a change to GitHub.

- [Using load balancers on Amazon Lightsail](#)

April 21, 2020 – How to load balance a simple Node.js web application using an Amazon Lightsail load balancer.

- [Building a photo diary on Amazon Lightsail with Ghost](#)

March 23, 2020 – Start a photo diary using Ghost on Lightsail.

- [Amazon Lightsail database tips and tricks](#)

March 23, 2020 – Use advanced features found in Amazon Relational Database Service (Amazon RDS).

- [Configuring and using monitoring and Notifications](#)

February 27, 2020 – Creating notification contacts, creating a new alarm, and testing out notifications with resource monitoring.

- [Deploying a highly-available WordPress site on Amazon Lightsail, Part 1: Implementing a highly-available Lightsail database with WordPress](#)

October 22, 2019 – Build a highly-available WordPress site on Lightsail, part 1.

- [Deploying a highly-available WordPress site on Amazon Lightsail, Part 2: Using Amazon S3 with WordPress to securely deliver media files](#)

October 31, 2019 – Build a highly-available WordPress site on Lightsail, part 2.

- [Deploying a highly-available WordPress site on Amazon Lightsail, Part 3: Increasing security and performance using Amazon CloudFront](#)

November 7, 2019 – Build a highly-available WordPress site on Lightsail, part 3.

- [Deploying a highly-available WordPress site on Amazon Lightsail, Part 4: Increasing performance and scalability with a Lightsail load balancer](#)

November 14, 2019 – Build a highly-available WordPress site on Lightsail, part 4.

- [Building a pocket platform-as-a-service with Amazon Lightsail](#)

October 8, 2019 – Assemble a pocket platform on Lightsail.

- [Deploying a Nginx-based HTTP/HTTPS load balancer with Amazon Lightsail](#)

July 8, 2019 – Set up a NGINX-based load balancer inside of a Lightsail instance.

- [New to the AWS Cloud? Amazon Lightsail can help](#)

March 27, 2019 – Getting started on Amazon Lightsail.

- [New – Managed databases for Amazon Lightsail](#)

October 16, 2018 – Create a managed database with a couple of clicks.

- [Amazon Lightsail update: More instance sizes and price reductions](#)

August 23, 2018 – Lightsail instance overview.

- [Amazon Lightsail: The power of AWS, the simplicity of a VPS](#)

November 30, 2016 – Lightsail launch announcement.

Tutorials

Top 5 hands-on tutorials:

1. [Create a load balanced WordPress website](#)

September 8, 2021 – Launch a highly available WordPress website with Lightsail.

2. [Migrating and managing a WordPress website with Amazon Lightsail](#)

February 22, 2021 – Launch a clone of your WordPress website onto Lightsail using the Seahorse software.

3. [Launch a Linux virtual machine](#)

September 11, 2020 – Launch, configure, and connect to a Linux instance with Lightsail.

4. [Launch a Windows virtual machine](#)

September 11, 2020 – Launch, configure, and connect to a Windows instance with Lightsail.

5. [Launch a cPanel and WHM instance on Amazon Lightsail](#)

July 27, 2020 – This tutorial walks through a few steps that you can take after your cPanel and WHM instance is up and running on Lightsail.

- [How to setup and configure Magento on Amazon Lightsail](#)

August 11, 2021 – Get an e-commerce site up and running.

- [How to connect your WordPress site to an object storage bucket](#)

July 14, 2021 – Set up your WordPress site on Lightsail and connect the website to a Lightsail bucket.

- [Create object storage buckets](#)

July 14, 2021 – Create an object storage bucket in Amazon Lightsail.

- [Connecting a WordPress website to an Amazon Lightsail bucket and distribution](#)

July 14, 2021 – Configure your Lightsail bucket as the origin of a Lightsail content delivery network (CDN) distribution.

- [How to setup and configure Plesk](#)

April 22, 2021 – Get a Plesk hosting stack up and running on Lightsail.

- [How to Setup a Prestashop e-commerce site](#)

April 1, 2021 – Launch and configure a Lightsail instance using the PrestaShop Certified by Bitnami blueprint.

- [How to Use Amazon EFS with Amazon Lightsail](#)

March 15, 2021 – Create and connect to an Amazon EFS file system from Lightsail instances using VPC peering.

- [How to setup a Nginx reverse proxy](#)

February 10, 2021 – Set up a Nginx reverse proxy using Lightsail containers.

- [How to Serve a Flask pp](#)

February 3, 2021 – Learn how to serve a Flask application with Lightsail containers.

- [Creating, pushing, and deploying container images with Amazon Lightsail](#)

November 11, 2020 – Create a container image on your local machine using a Dockerfile.

- [Build a Drupal website](#)

September 11, 2020 – Deploy and host a production-ready Drupal website on Lightsail.

- [Build a LAMP stack web App](#)

September 9, 2020 – Launch and run a highly available PHP web application on Lightsail.

- [Configure your WordPress instance to work with your distribution](#)

July 16, 2020 – Configure your WordPress instance to work with your Lightsail distribution.

- [Launch a WordPress website](#)

March 23, 2020 – Get a website up and running with WordPress installed on a Lightsail virtual machine.

- [Host a .NET application](#)

March 20, 2020 – Build and deploy a .NET application using Lightsail.

- [Map your domain at Amazon Route 53 to your Lightsail resources](#)

Route traffic for your domain, such as example.com, to your Lightsail resources.

Videos

- [Amazon Lightsail Tutorial: Deploy a Django app](#)

July 14, 2021 – In this tutorial, you create a Django application.

- [Amazon Lightsail Tutorial: Deploy a Flask app](#)

July 14, 2021 – In this tutorial, you create a Flask application.

- [Amazon Lightsail Tutorial: Deploy a NGINX reverse proxy](#)

July 14, 2021 – Create a Flask application, build a Docker container, create a container service on Lightsail, and then deploy the application.

- [Amazon Lightsail Tutorial: Deploy an e-commerce site](#)

July 14, 2021 – Launch a Lightsail instance using the PrestaShop Certified by Bitnami blueprint, and configure it.

- [Deploy a containerized application on Amazon Lightsail](#)

December 29, 2020 – Learn how to deploy a containerized application in Lightsail.

- [Amazon Lightsail Tutorial: Build a Drupal website](#)

August 31, 2020 – Launch and configure a Drupal instance.

- [Amazon Lightsail Tutorial: Deploy a LAMP Stack app](#)

August 31, 2020 – Deploy a LAMP (Linux Apache MySQL PHP) stack application onto a single Lightsail instance.

- [Amazon Lightsail Tutorial: Launch a Linux instance](#)

August 31, 2020 – Learn how to launch a Linux instance.

- [Amazon Lightsail Tutorial: Launch a Windows instance](#)

August 31, 2020 – Learn how to launch a Windows instance.

- [Amazon Lightsail Tutorial: Run your own Minecraft server](#)

August 31, 2020 – Learn how to set up a dedicated Minecraft server.

- [Introduction to Amazon Lightsail tutorials](#)

August 31, 2020 – Get started on your cloud journey today with Lightsail.

- [Amazon Lightsail: The easiest way to get started on AWS](#)

March 20, 2020 – Lightsail is the easiest way to get started on AWS. It offers virtual servers, storage, databases and networking, plus a cost-effective, monthly plan.

- [Configuring a Plesk instance in Amazon Lightsail](#)

March 27, 2019 – Learn how to configure a Plesk instance in Lightsail.

- [Configuring WordPress Multisite in Amazon Lightsail](#)

January 15, 2019 – Learn how to configure a WordPress Multisite instance in Lightsail.

- [Managing Lightsail](#)

October 9, 2018 – Take a quick look at Lightsail key features.

- [Deploy a MEAN stack app on Amazon Lightsail](#)

June 5, 2018 – Use Lightsail's MEAN blueprint to deploy a custom application to the cloud.

- [Deploy a WordPress instance on Amazon Lightsail](#)

June 5, 2018 – Deploy a WordPress instance on Lightsail.

Tutorial: Migrate data from a MySQL 5.6 database to a newer database version

In this tutorial, we show you how to migrate data from a MySQL 5.6 database to a new MySQL 5.7 database in Amazon Lightsail. To perform the migration, you connect to your MySQL 5.6 database and export the existing data. You then connect to the MySQL 5.7 database and import the data. After the new database has the required data, you can reconfigure your application to connect to the new database.

Contents

- [Step 1: Understand the changes](#)
- [Step 2: Complete the prerequisites](#)
- [Step 3: Connect to your MySQL 5.6 database and export the data](#)
- [Step 4: Connect to your MySQL 5.7 database and import the data](#)
- [Step 5: Test your application and complete the migration](#)

Step 1: Understand the changes

Going from a MySQL 5.6 database to a MySQL 5.7 database is considered a major version upgrade. Major version upgrades can contain database changes that are not backward-compatible with existing applications. We recommend that you thoroughly test any upgrade before applying it to your production instances. For more information, see [Changes in MySQL 5.7](#) in the *MySQL documentation*.

We recommend that you first migrate your data from your existing MySQL 5.6 database to a new MySQL 5.7 database. Then test your application with your new MySQL 5.7 database on a pre-production instance. If your application behaves as expected, apply the change to your application in the production instance. To take it a step further, you can then migrate your data from your existing MySQL 5.7 database to a new MySQL 8.0 database, test your application in pre-production again, and apply the change to your application in production.

Step 2: Complete the prerequisites

You must complete the following prerequisites before continuing to the next sections of this tutorial:

- Install MySQL Workbench on your local computer, which you will use to connect to your databases to export and import data. For more information, see [MySQL Workbench download](#) on the *MySQL website*.
- Create a MySQL 5.7 database in Lightsail. For more information, see [Creating a database in Amazon Lightsail](#).
- Enable public mode for your databases. This allows you to connect to them using MySQL Workbench. When you're done exporting and importing data, you can disable public mode for your databases. For more information, see [Configure the public mode for your database](#).
- Configure your MySQL Workbench to connect to your databases. For more information, see [Connect to your MySQL database](#).

Step 3: Connect to your MySQL 5.6 database and export the data

In this section of the tutorial, you will connect to your MySQL 5.6 database and export data from it using MySQL Workbench. For more information about using MySQL Workbench to export data, see [SQL Data Export and Import Wizard](#) on the *MySQL Workbench Manual*.

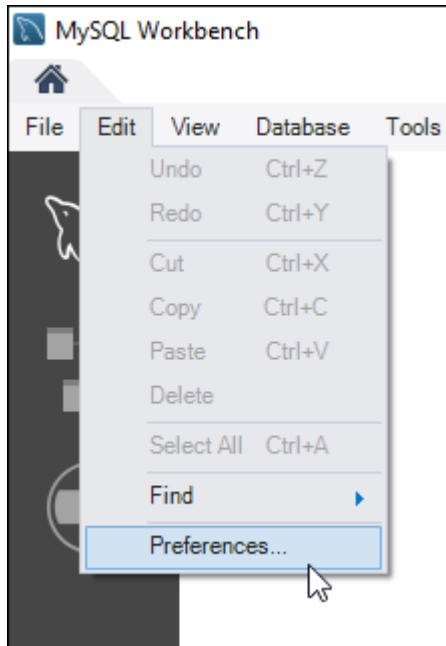
1. Connect to your MySQL 5.6 database using MySQL Workbench.

MySQL Workbench uses `mysqldump` to export data. The version of `mysqldump` used by MySQL Workbench must be the same (or later) as the version of the MySQL database from which you will export data. For example, if you're exporting data from a MySQL 5.6.51 database, then you must use `mysqldump` version 5.6.51 or later. You might need to download and install the appropriate version of MySQL server on your local computer in order to ensure you're using the correct version of `mysqldump`. To download a specific version of MySQL server, see [MySQL](#)

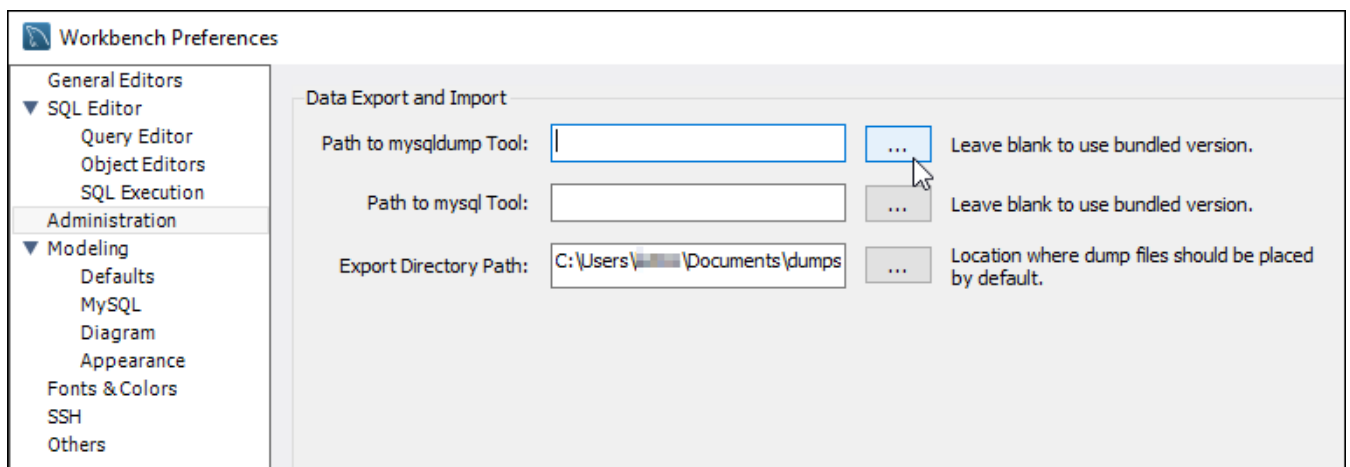
[Community Downloads](#) on the *MySQL website*. The **MySQL Installer for Windows MSI** offers the option to download any version of MySQL server.

Complete the following steps to choose the correct version of `mysqldump` to use in MySQL Workbench:

1. In MySQL Workbench, choose **Edit**, and then choose **Preferences**.

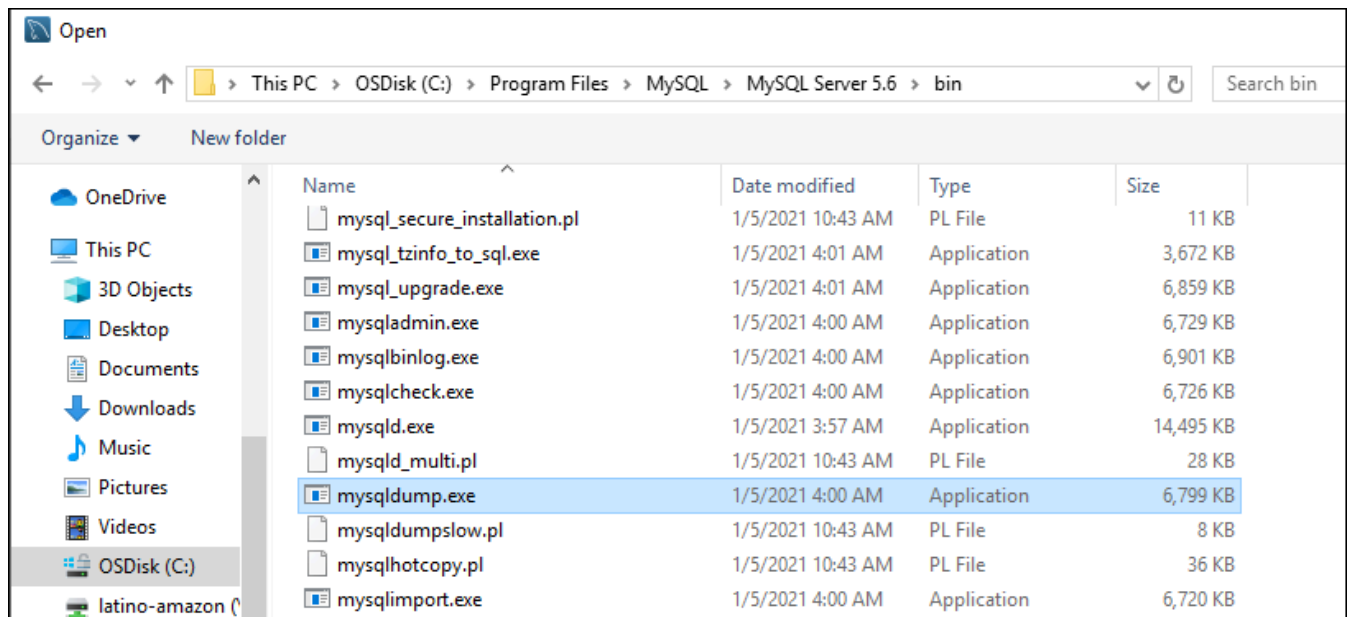


2. Choose **Administration** in the navigation pane.
3. In the **Workbench Preferences** window that appears, choose the ellipsis button next to the **Path to mysqldump Tool** text box.

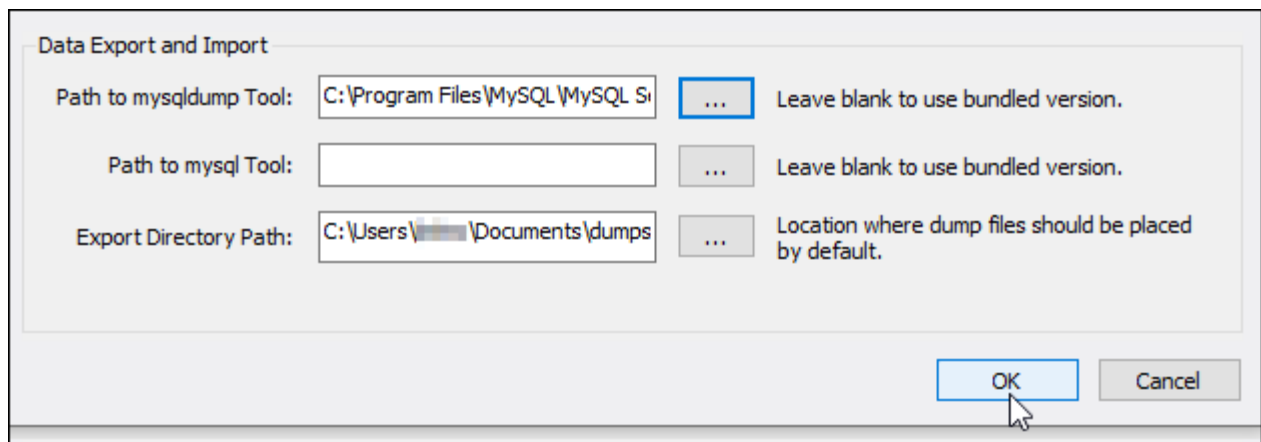


4. Browse to the location of the appropriate `mysqldump` executable file, and double-click it.

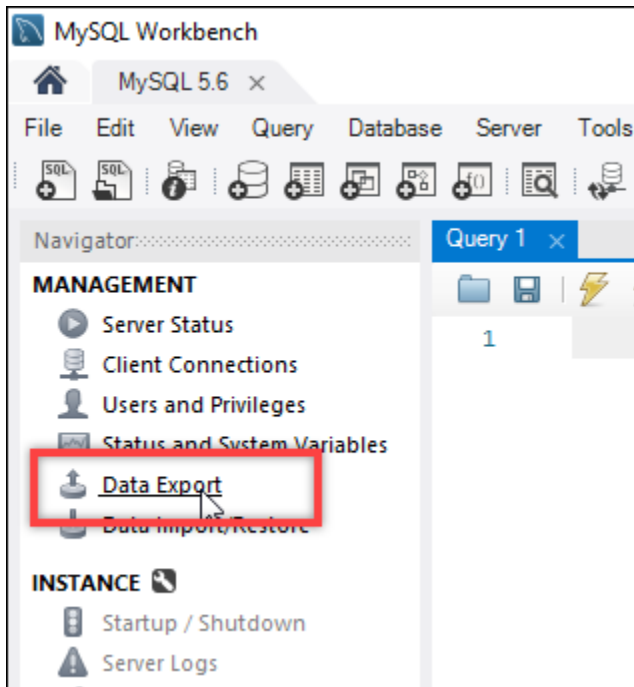
In Windows, the `mysqldump.exe` file is typically located in the `C:\Program Files\MySQL\MySQL Server 5.6\bin` directory. In Linux, enter `which mysqldump` in the terminal to see where the **mysqldump** file is located.



5. Choose **OK** in the in the **Workbench Preferences window**.



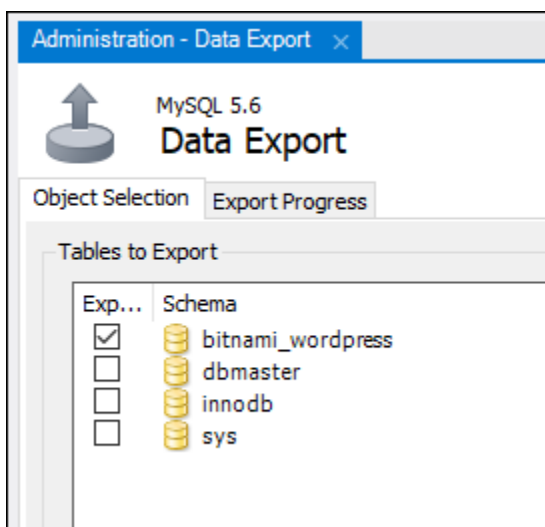
2. Choose **Data Export** in the **Navigator** pane



3. In the **Data Export** tab that appears, add a check mark next to the tables that you wish to export.

Note

In this example, we chose the `bitnami_wordpress` table that contains data for a WordPress website on a "Certified by Bitnami" WordPress instance.



- In the **Export Options** section, choose **Export to Self-Contained File**, and then make a note of the directory in which the export file will be saved.

Export Options

Export to Dump Project Folder

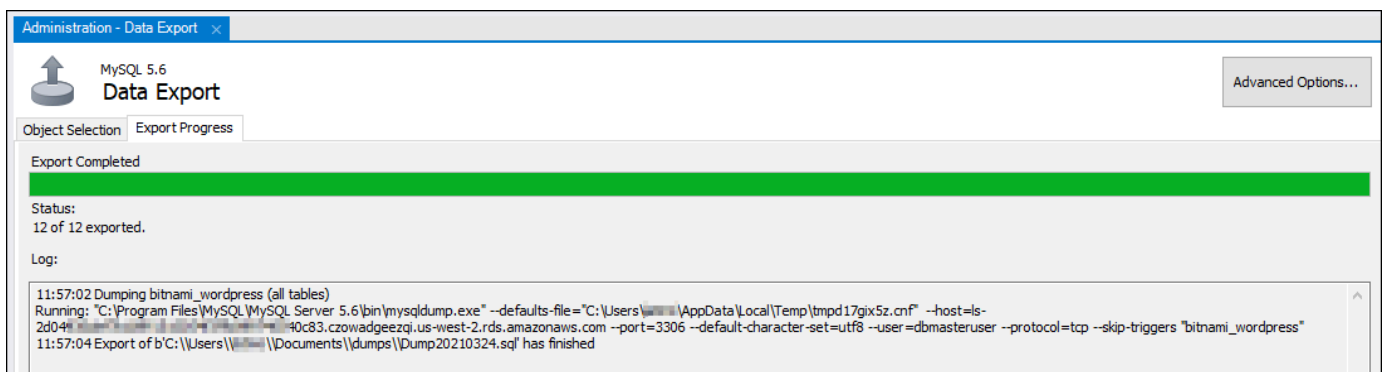
Each table will be exported into a separate file. This allows a selective restore, but may be slower.

Export to Self-Contained File

All selected database objects will be exported into a single, self-contained file.

Create Dump in a Single Transaction (self-contained file only) Include Create Schema

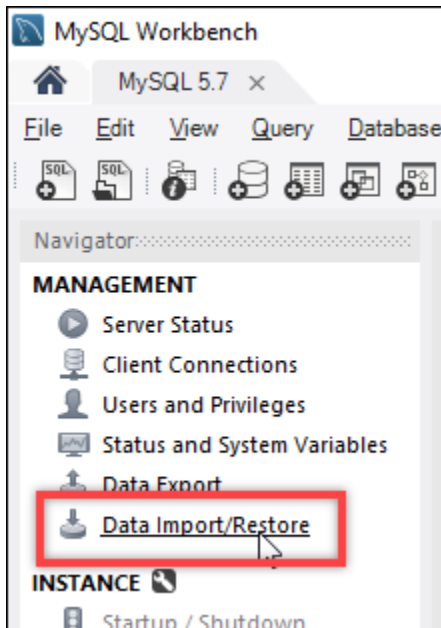
- Choose **Start Export**.
- Wait for the export to complete before continuing to the next section of this tutorial.



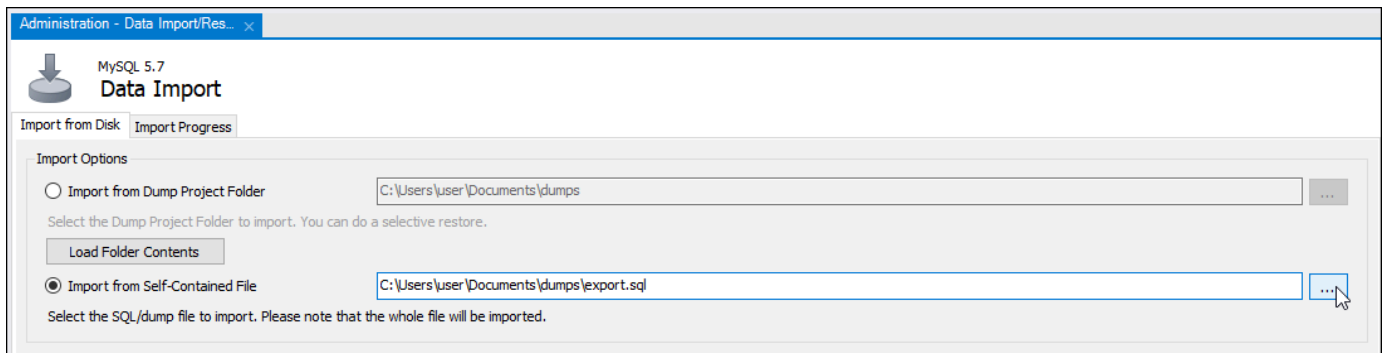
Step 4: Connect to your MySQL 5.7 database and import the data

In this section of the tutorial, you will connect to your MySQL 5.7 database and import data to it using MySQL Workbench.

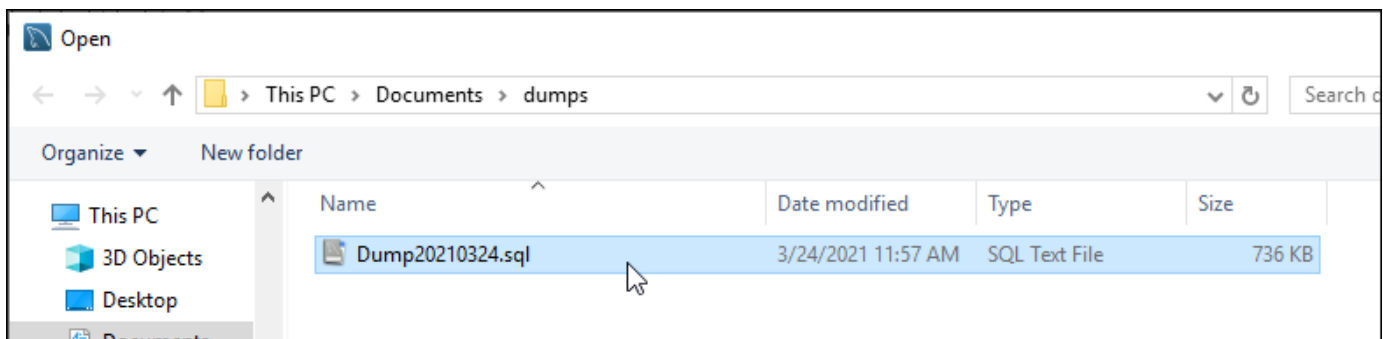
- Connect to your MySQL 5.7 database using MySQL Workbench on your local computer.
- Choose **Data Import/Restore** in the **Navigator** pane.



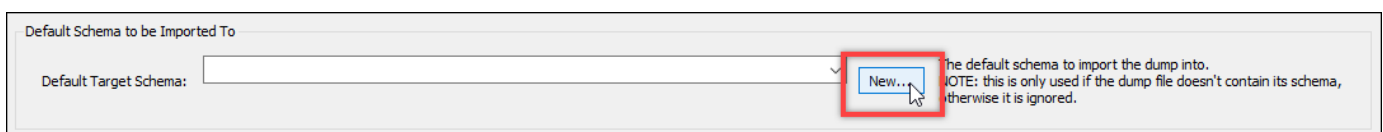
3. In the **Data Import** tab that appears, choose **Import from Self-Contained File**, and then choose the ellipsis button next to the text box.



4. Browse to the location where the export file was saved, and double-click it.



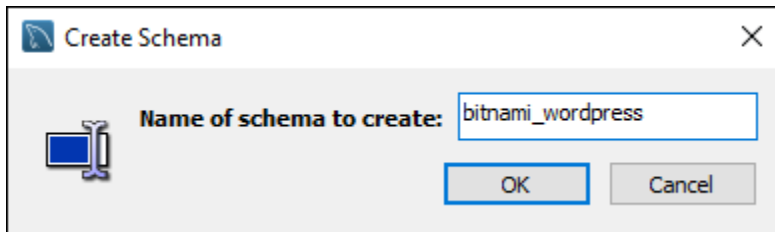
5. Choose **New** in the **Default Schema to be Imported To** section.



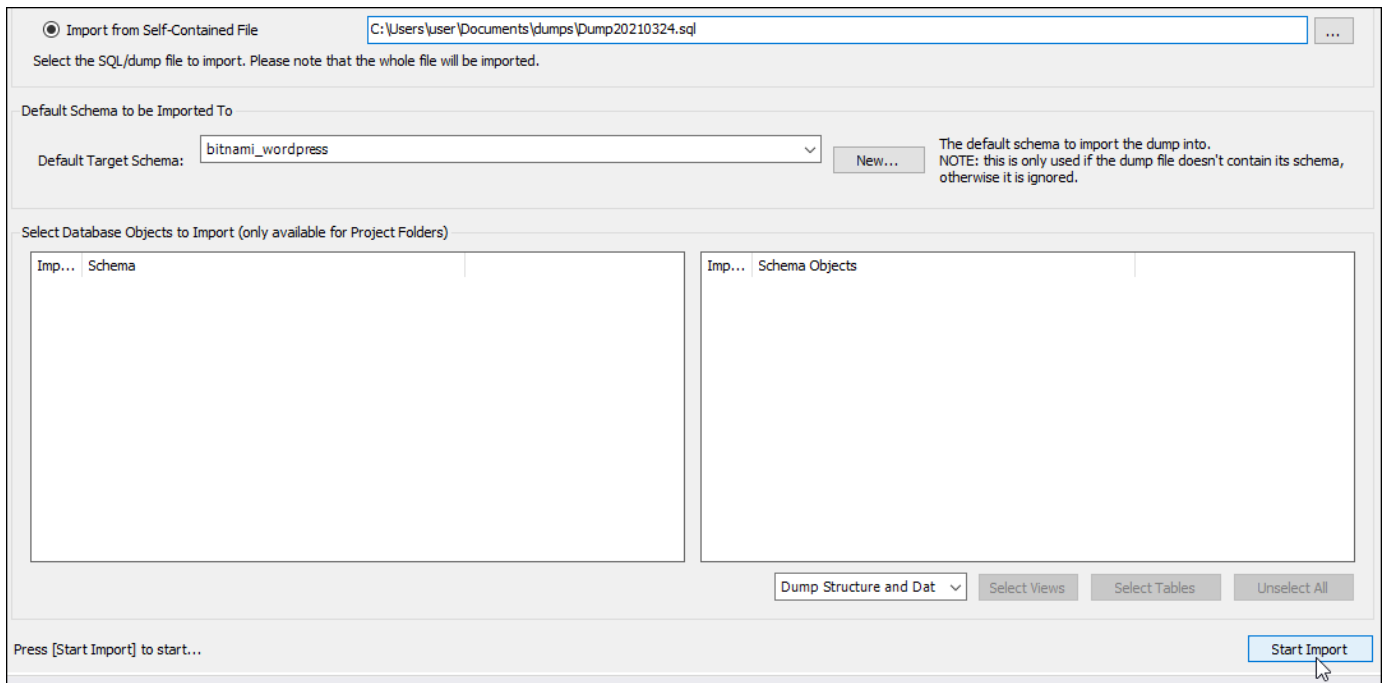
6. Enter the name of the schema in the **Create Schema** window that appears.

Note

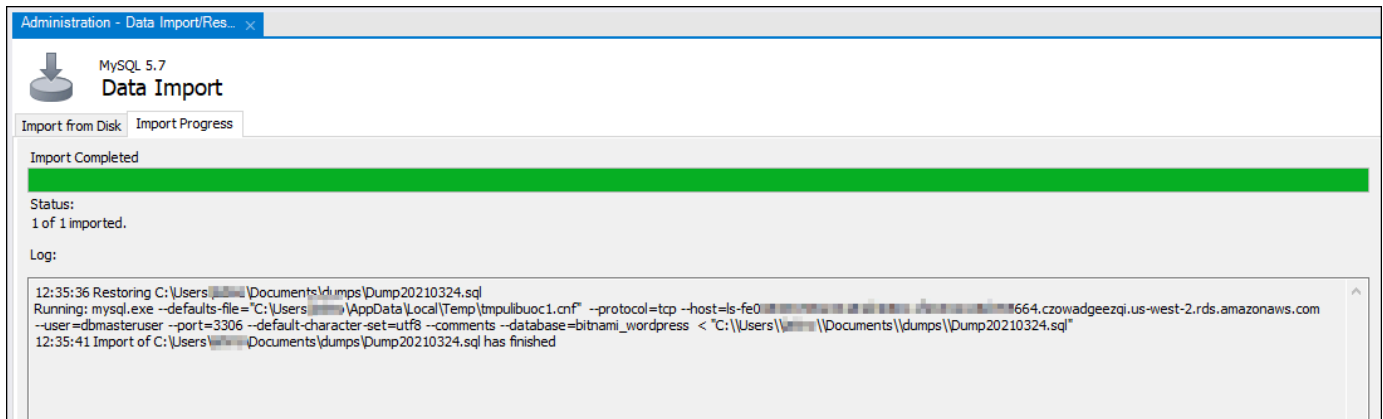
In this example, we enter `bitnami_wordpress` because that is the name of the database table that we exported.



7. Choose **Start Import**.



8. Wait for the import to complete before continuing to the next section of this tutorial.



Step 5: Test your application and complete the migration

At this point, your data is now in your new MySQL 5.7 database. Configure your application in a pre-production environment, and test the connection between your application and your new MySQL 5.7 database. If your application behaves as expected, then proceed to make the change to your application in the production environment.

When you're finished with the migration, you should disable the public mode for your databases. You can delete your MySQL 5.6 database when you are certain you no longer need it. However, you should create a snapshot of your MySQL 5.6 database before you delete it. While you're at it, you should also create a snapshot of your new MySQL 5.7 database. For more information, see [Create a database snapshot](#).

Set up and configure Plesk in Lightsail

You can create a Plesk hosting stack in Amazon Lightsail that includes the following features.

- WordPress Toolkit, featuring automation in a graphical user interface
- Let's Encrypt support for SSL certificates and configuring encrypted (HTTPS) traffic on a single instance
- FTP access to transfer files to and from your instance
- Docker Proxy Rules
- Web-based server management and security tools, including Plesk Firewall, Logs, and ModSecurity

This guide shows you how to create a Plesk instance in Lightsail, and how to sign in to the Plesk panel for the first time by creating a username and password.

⚠ Important

If you experience issues after launching your Plesk instance, go to the Plesk support page to see if there are updates that need to be installed on the instance. For more information, see the [Plesk help center](#) and [Plesk Updates](#) in the *Plesk Documentation and Help Portal*.

Create a Plesk instance

Complete the following steps to create a Plesk instance in Lightsail.

1. Sign in to the Lightsail console at <https://lightsail.aws.amazon.com/>.
2. In the **Instances** tab of the Lightsail home page, choose **Create instance**.
3. Choose the location where you want to create your instance.

Choose **Change AWS Region and Availability Zone** to change your instance location.

4. Under **Apps + OS**, choose **Plesk Hosting Stack on Ubuntu**.
5. Choose your instance plan.

ℹ Note

Plesk is not supported on the \$3.50 USD per month Lightsail plan.

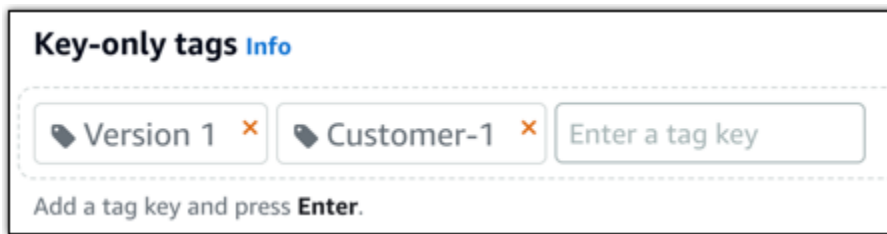
6. Enter a name for your instance.

Resource names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2 to 255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

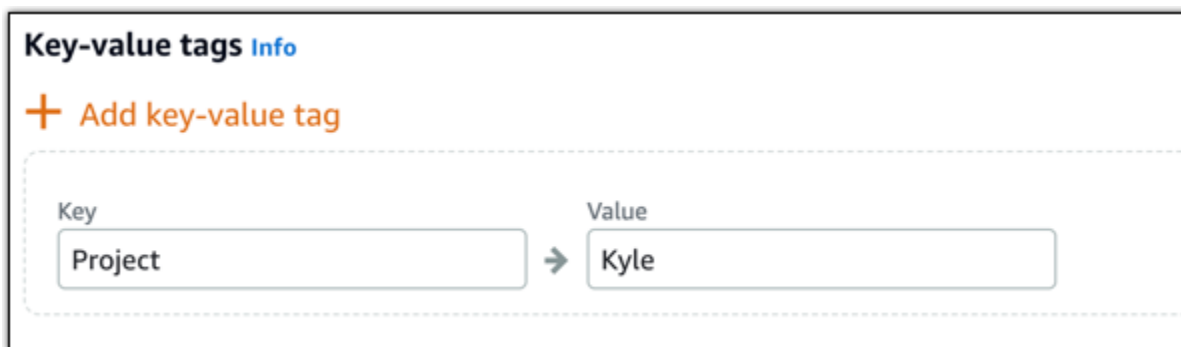
7. Choose one of the following options to add tags to your instance:

- **Add key-only tags** or **Edit key-only tags** (if tags have already been added). Enter your new tag into the tag key text box, and press **Enter**. Choose **Save** when you're done entering your tags to add them, or choose **Cancel** to not add them.



- **Create a key-value tag**, then enter a key into the **Key** text box, and a value into the **Value** text box. Choose **Save** when you're done entering your tags, or choose **Cancel** to not add them.

Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Note

For more information about key-only and key-value tags, see [Tags](#).

8. Choose **Create instance**.

The instance requires a few minutes to provision and become available after you create it.

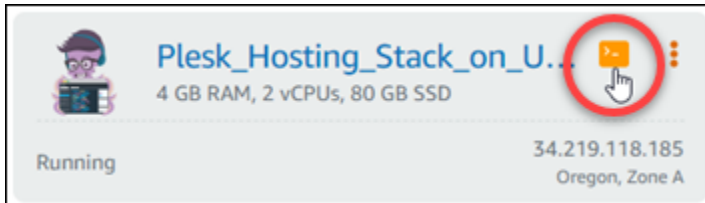
Note

If you want to use Plesk on Amazon Lightsail for web hosting, you should [attach a static IP address to your instance](#). If you attach a static IP, you will need to reboot your instance in Lightsail before you can log into it for the first time.

Configure a username and password for your Plesk instance

Complete the following steps to configure a username and password for your Plesk instance, and sign in to the Plesk panel for the first time.

1. In the **Instances** tab of the Lightsail home page, choose the SSH quick connect icon for the Plesk instance that you want to set up.



2. Enter the following command.

```
sudo plesk login | grep -v internal:8
```

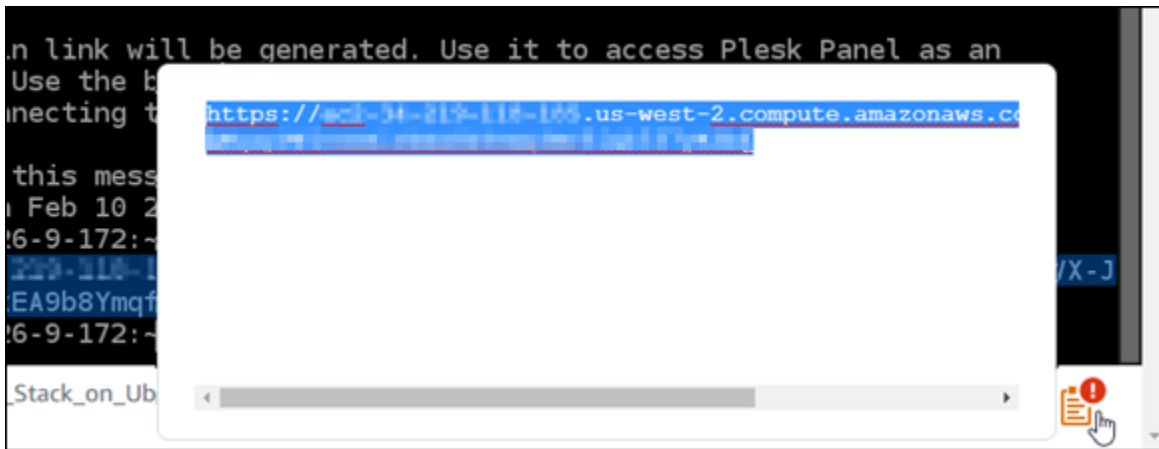
You should see a result similar to the following example.

```
ubuntu@ip-10-10-10-10:~$ sudo plesk login
https://34.219.118.185.us-west-2.compute.amazonaws.com/login?secret=VFmhiq5NSN81d-Ebn
https://34.219.118.185/login?secret=VFmhiq5NSN81d-Ebn
ubuntu@ip-10-10-10-10:~$
```

⚠ Important

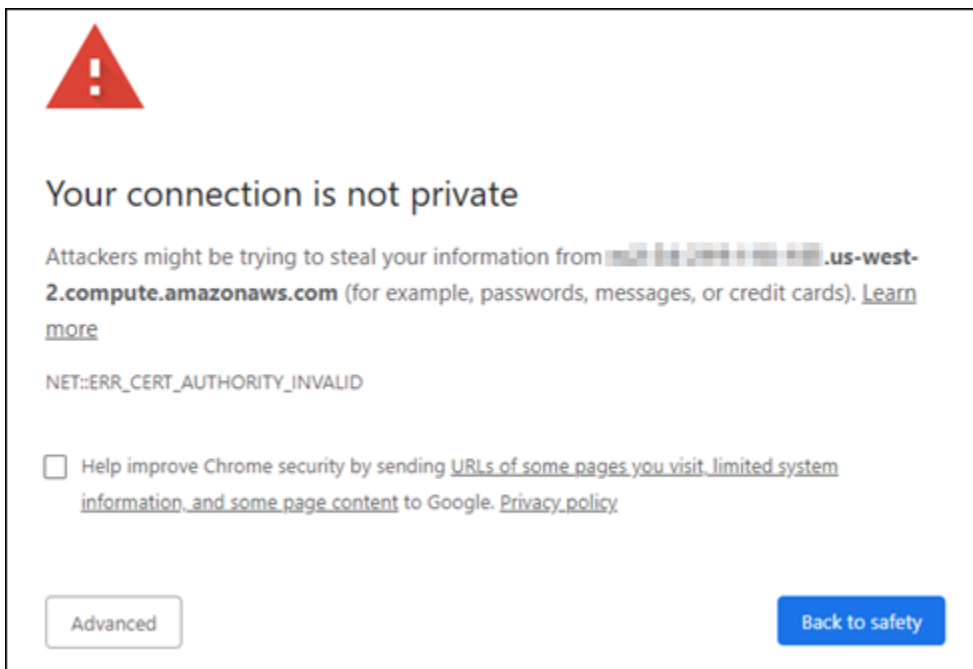
If you recently attached a static IP to your Plesk instance, you might get a one-time login URL that uses the old public IP address. Reboot the instance, and run the above command again to get a one-time login URL that uses the new static IP address.

3. Highlight the URL shown in the browser-based SSH window, then choose the clipboard icon, and copy the URL to your local clipboard.



4. Open a new browser window, and browse to the URL that you copied.

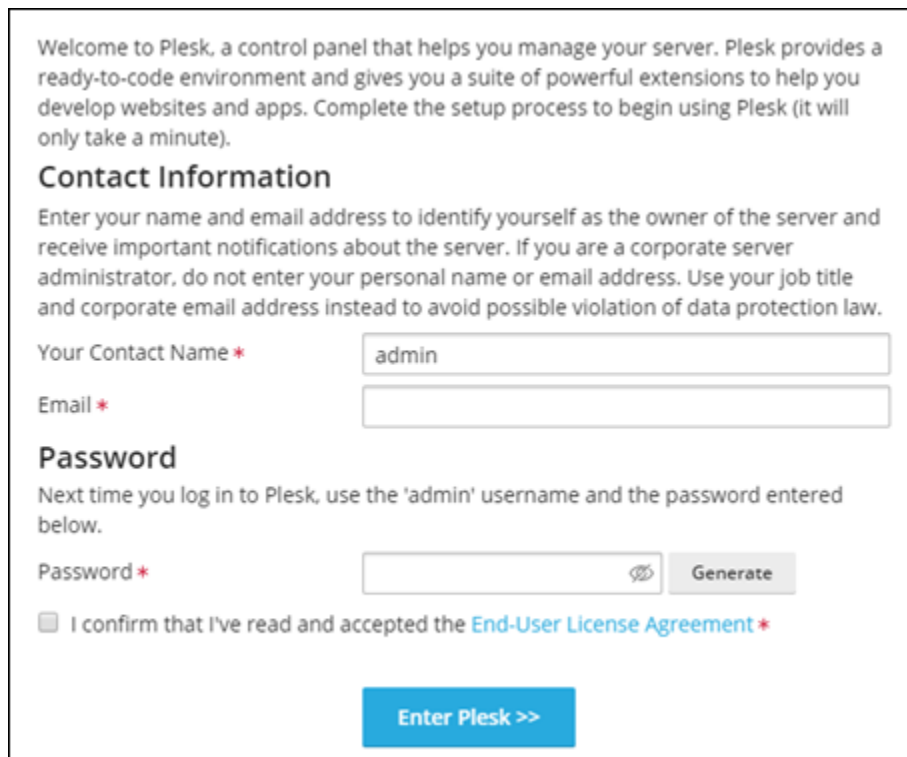
You might see a browser warning that your connection is not private, not secure, or that there's a security risk. This happens because your Plesk instance does not yet have an SSL/TLS certificate applied to it. The prompt may be different than what is shown in the following example depending on the browser you use.



5. Complete one of the following steps depending on the browser that you use:
 - **Chrome** — Choose **Advanced**, and then choose **Proceed** to continue to the Plesk set up page.
 - **Edge** — Choose **Details**, and then choose **Go on to the webpage (Not recommended)** to continue to the Plesk set up page.

- **Firefox** — Choose **Advanced**, and then choose **Accept the Risk and Continue** to continue to the Plesk set up page.
 - **Internet Explorer** — Choose **More information**, and then choose **Go on to the webpage (Not recommended)** to continue to the Plesk set up page.
6. Enter your contact name, email address, and password.

In this page, you can change the default admin contact name if you prefer to use something different. However, that is only the display name; your username to sign in to Plesk will continue to be admin.



Welcome to Plesk, a control panel that helps you manage your server. Plesk provides a ready-to-code environment and gives you a suite of powerful extensions to help you develop websites and apps. Complete the setup process to begin using Plesk (it will only take a minute).

Contact Information

Enter your name and email address to identify yourself as the owner of the server and receive important notifications about the server. If you are a corporate server administrator, do not enter your personal name or email address. Use your job title and corporate email address instead to avoid possible violation of data protection law.

Your Contact Name *

Email *

Password

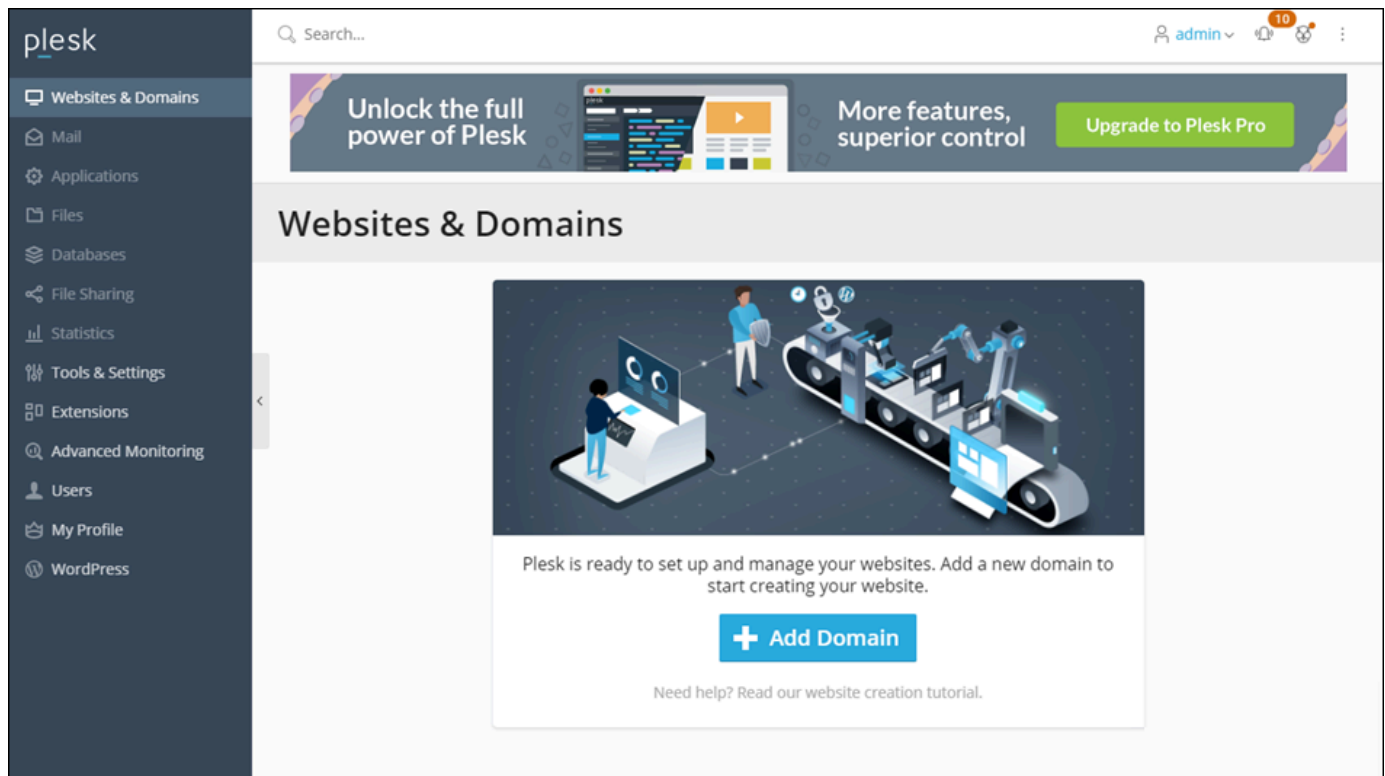
Next time you log in to Plesk, use the 'admin' username and the password entered below.

Password *

I confirm that I've read and accepted the [End-User License Agreement](#) *

7. Confirm that you accept the end-user license agreement, and choose **Enter Plesk**.

If successful, you will be signed in to the Plesk panel where you can add your domain, and begin managing your websites.

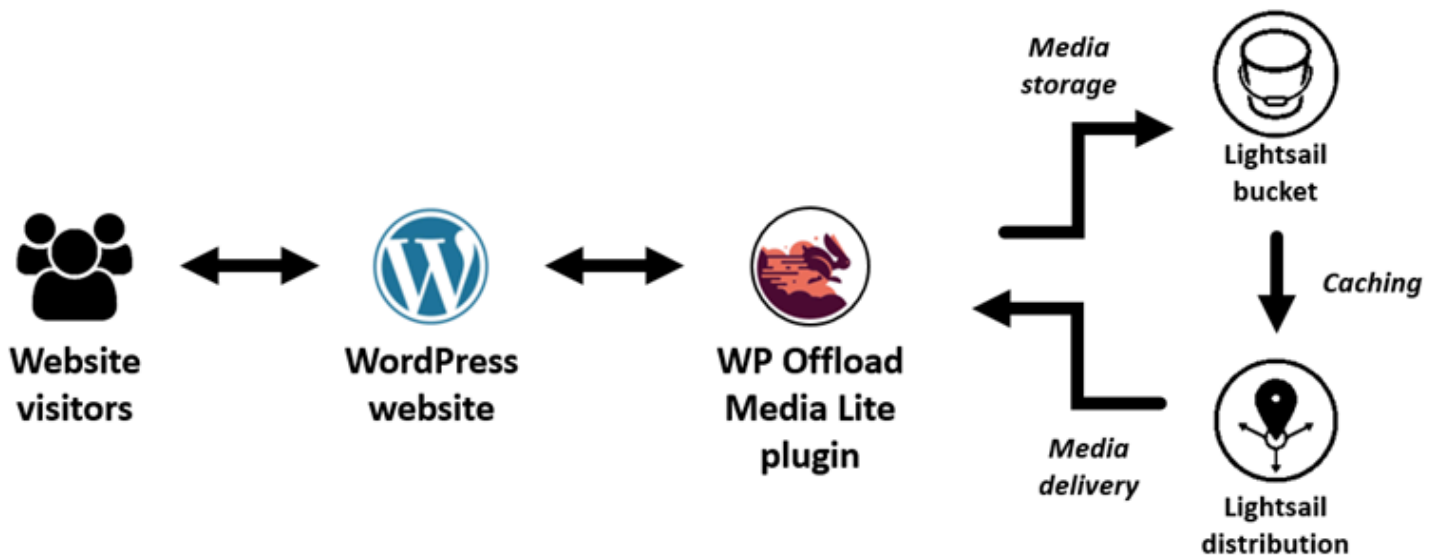


If you need to sign in again later, just navigate to `https://PublicIPAddress:8443`. Replace *PublicIPAddress* with the public IP address or static IP address of your instance. For example, `https://192.0.2.0/:8443`. Then enter the username and password you created earlier to sign in to the Plesk panel.

For more information about using Plesk, see [Getting Started with Managing Websites in Plesk](#) in the *Plesk Documentation and Help Portal*.

Tutorial: Use a Lightsail bucket with a content delivery network distribution

This tutorial describes the steps required to configure your Amazon Lightsail bucket as the origin of a Lightsail content delivery network (CDN) distribution. It also describes how to configure your WordPress website to upload and store media (such as images and movies files) on your bucket, and deliver media from your distribution. One example of how to do this is with the [WP Offload Media Lite plugin](#). The following diagram illustrates this configuration.



Storing website media in a Lightsail bucket takes the load off your instance from having to store and serve those files. Caching and serving media from a Lightsail distribution speeds up the delivery of those files to your website visitors, and can improve overall website performance. For more information about distributions, see [Content delivery network distributions](#). For more information about buckets, see [Object storage](#).

Contents

- [Step 1: Complete the prerequisites](#)
- [Step 2: Modify your bucket permissions](#)
- [Step 3: Create a distribution with a bucket as the origin](#)
- [Step 4: Enable a custom subdomain for your distribution](#)
- [Step 5: Install the WP Offload Media Lite plugin on your WordPress website](#)
- [Step 6: Test the connection between your WordPress website and your Lightsail bucket and distribution](#)

Step 1: Complete the prerequisites

Complete the following prerequisites if you haven't already:

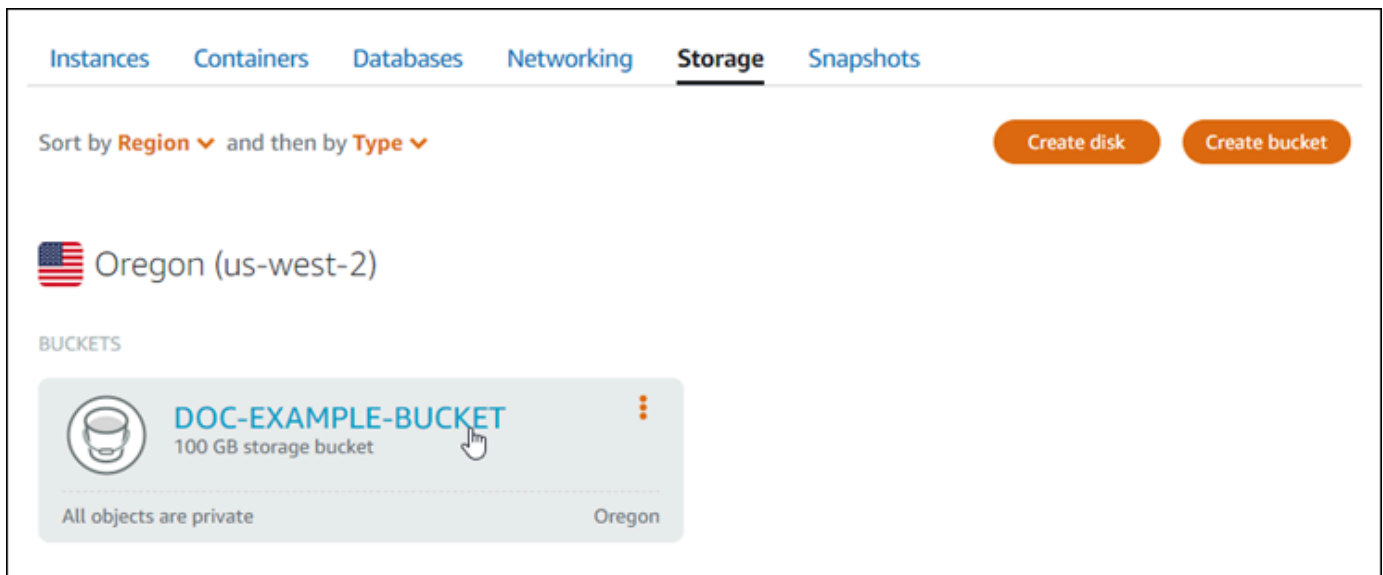
- Create and configure a WordPress instance in Lightsail, and get the password to sign in to the administration dashboard. For more information, see [Tutorial: Launch and configure a WordPress instance in Amazon Lightsail](#).

- Create a bucket in the Lightsail object storage service. For more information, see [Creating buckets in Lightsail](#).

Step 2: Modify your bucket permissions

Complete the following procedure to give your WordPress instance and the WP Offload Media Lite plugin access to your bucket. The permissions of your bucket must be set to **Individual objects can be made public (read only)**. You must also attach your WordPress instance to your bucket. For more information about bucket permissions, see [Bucket permissions](#).

1. Sign in to the [Lightsail console](#).
2. On the Lightsail home page, choose the **Storage** tab.
3. Choose the name of the bucket that you want to use with your WordPress website.



4. Choose the **Permissions** tab on the **Bucket management** page.
5. Choose **Change permissions** under the **Bucket access permissions** section of the page.


[Objects](#) **Permissions** [Metrics](#) [Versioning](#)

Bucket access permissions

Manage the anonymous access to objects in this bucket. You can make all objects **private** or **public (read-only)**. Alternatively, you can keep your bucket private while making individual objects public (read-only).

[Learn more about bucket permissions](#)

[Change permissions](#)

 **All objects are private**
Your objects are readable only by you or anyone you give access to.

Programmatic access

Programmatic access gives plugins, instances, and other resources full access to this bucket and its objects. You can grant programmatic access by using either of


6. Choose **Individual objects can be made public and read only**.


Bucket access permissions


Manage the anonymous access to objects in this bucket. You can make all objects **private** or **public (read-only)**. Alternatively, you can keep your bucket private while making individual objects public (read-only).

[Learn more about bucket permissions](#)

[Change permissions](#)

 **All objects are private**
Your objects are readable only by you or anyone you give access to.

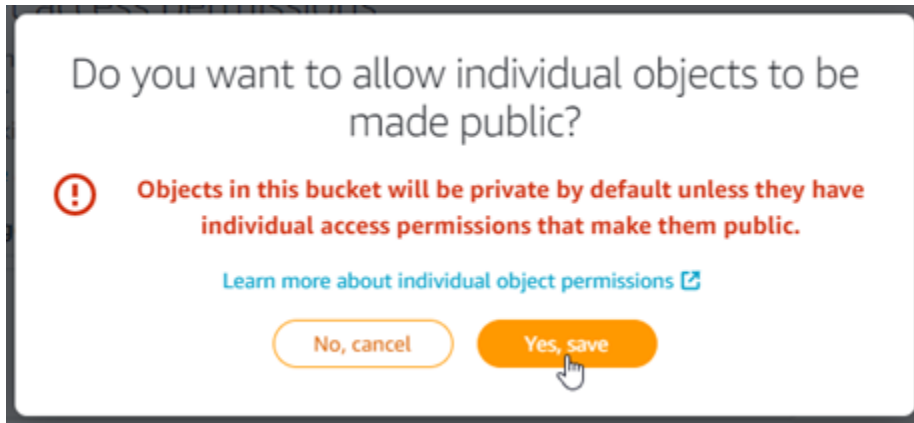
 **Individual objects can be made public (read-only)**
Your objects are readable only by you or anyone you give access to. But you can make individual objects readable by anyone in the world. Objects are private by default.

 **All objects are public (read-only)**
Your objects are public (read-only) by anyone in the world.

[Cancel](#) [Save](#)

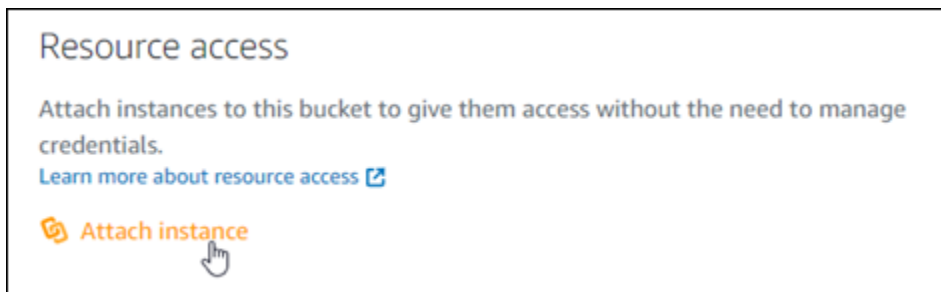
7. Choose **Save**.

8. Choose **Yes, save** in the confirmation prompt that appears.

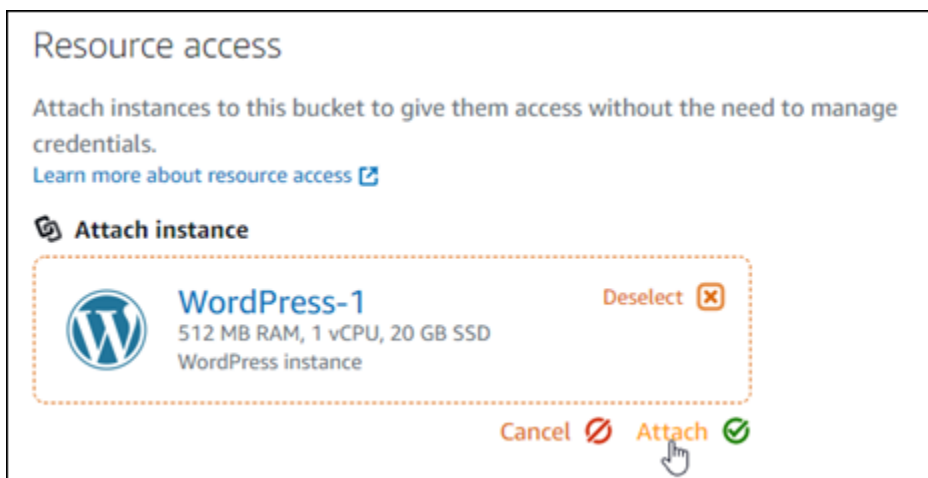


After a few moments, your bucket will be configured to allow for individual object access. This ensures that objects uploaded to your bucket from your WordPress website using the Offload Media Lite plugin are readable to your customers.

9. Scroll to the **Resource access** section of the page, and choose **Attach instance**.



10. Choose the name of your WordPress instance in the drop-down that appears, and then choose **Attach**.

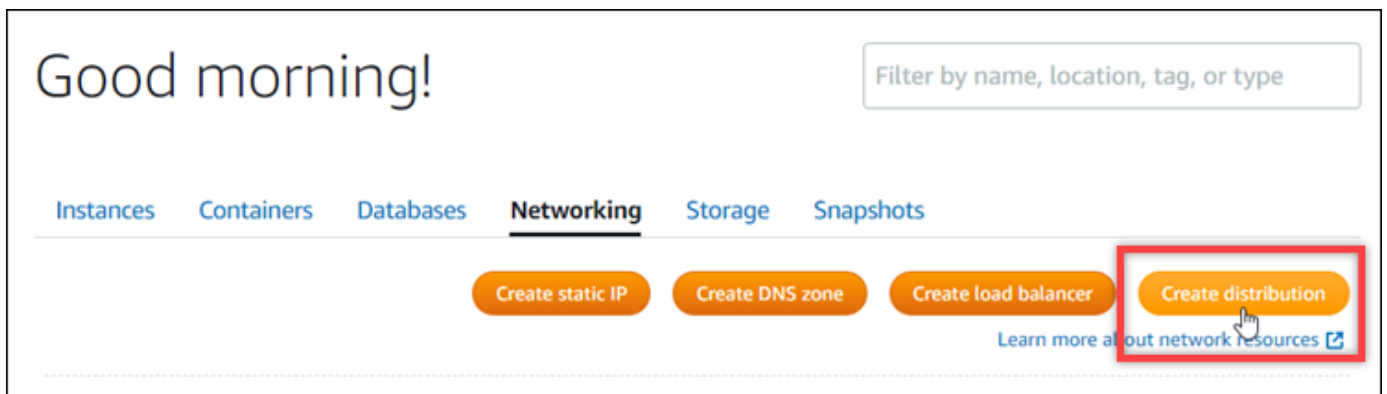


After a few moments, your WordPress instance is attached to your bucket. This gives your WordPress instance access to manage your bucket and its objects.

Step 3: Create a distribution with a bucket as the origin

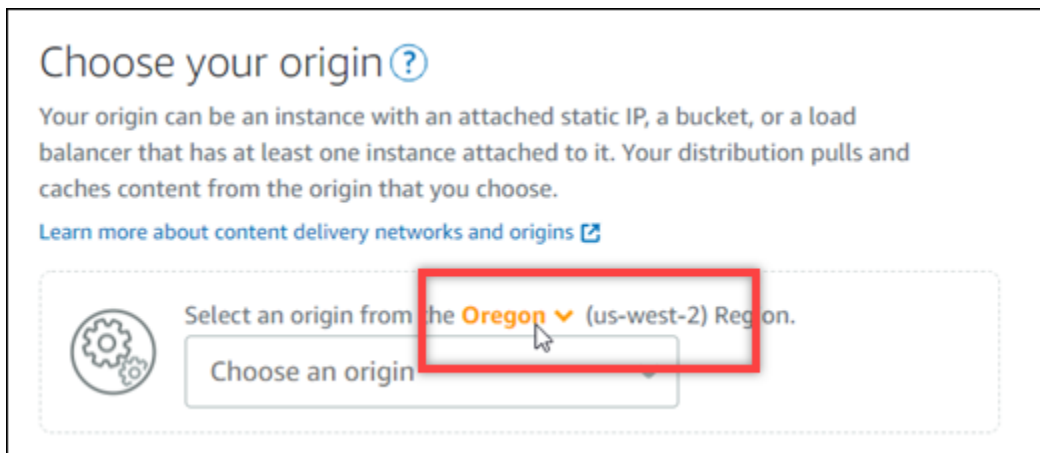
Complete the following procedure to create a Lightsail distribution and choose your Lightsail bucket as the origin.

1. Choose **Home** on the top navigation menu of the Lightsail console.
2. On the Lightsail home page, choose the **Networking** tab.
3. Choose **Create distribution**.

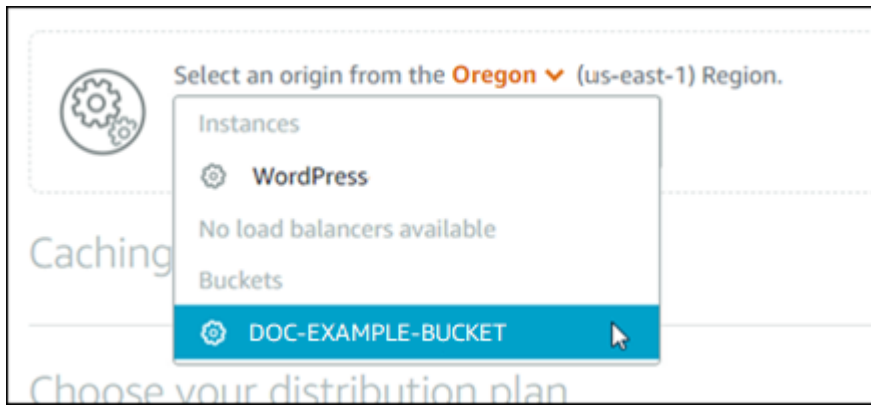


4. In the **Choose your origin** section of the page, choose the AWS Region in which you created your bucket.

Distributions are global resources. They can reference a bucket in any AWS Region, and distribute its content globally.



5. Choose your bucket as the origin.



Note

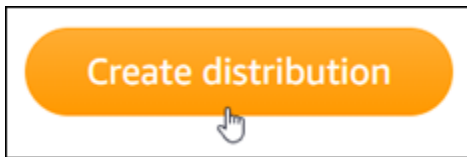
The permissions of your bucket must be set to **Individual objects can be made public (read only)**. Only individual objects that are public will be cached and served by the distribution. When you choose a bucket as the origin of a distribution, the options to specify the origin protocol policy, caching behavior, default behavior, and directory and file overrides become unavailable and cannot be edited. The origin protocol policy defaults to **HTTP only** for buckets, and the caching behavior defaults to **Cache everything**. You can change the advanced cache settings of the distribution after it's created.

6. Choose your distribution plan.
7. Enter a name for your distribution.

Distribution names:

- Must be unique within each AWS Region in your Lightsail account.
- Must contain 2-255 characters.
- Must start and end with an alphanumeric character or number.
- Can include alphanumeric characters, numbers, periods, dashes, and underscores.

8. Choose **Create distribution**.



Your distribution is created after a few moments. When your new distribution reaches an **Enabled** state, it is ready to serve and cache objects that are in your bucket.

Step 4: Enable a custom subdomain for your distribution

When you create your distribution, it is configured with a default domain that is similar to `123abc.cloudfront.net`. You can specify that default domain as the source of your media files when you configure the WP Offload Media Lite plugin. But we highly recommend that you enable a custom domain for your distribution. The custom domain that you enable for your distribution should be a subdomain of the domain that you're using with your WordPress website. For example, if you're using `mycustomdomain.com` with your WordPress website, then you might choose to use the custom domain `media.mycustomdomain.com` with your distribution. Using the same domain and subdomain combination between your WordPress website and your distribution helps improve the search engine optimization score of your website.

Complete the following steps to configure a custom domain for your distribution:

1. Create a Lightsail SSL/TLS certificate for your domain to use it with your distribution. Lightsail distributions require HTTPS, so you must request an SSL/TLS certificate for your domain before you can use it with your distribution. For more information, see [Create SSL/TLS certificates for your distribution](#).
2. Enable custom domains for your distribution to use your domain with your distribution. Enabling custom domains requires that you specify the Lightsail SSL/TLS certificate that you created for your domain. This adds your domain to your distribution and enables HTTPS. For more information, see [Enable custom domains for your distribution](#).
3. Add an alias record to your domain's DNS. After you add the alias record, users who visit your domain are routed through your distribution. For more information, see [Point your domain to a distribution](#).

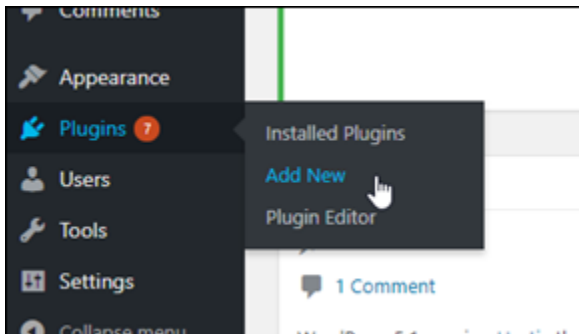
Step 5: Install the WP Offload Media Lite plugin on your WordPress website

Complete the following procedure to install the WP Offload Media Lite plugin on your WordPress website. This plugin automatically copies images, videos, documents, and any other media added through WordPress' media uploader to your Lightsail bucket. It can also be configured to serve media from your bucket through your Lightsail distribution. For more information, see [WP Offload Media Lite](#) in the *WordPress website*.

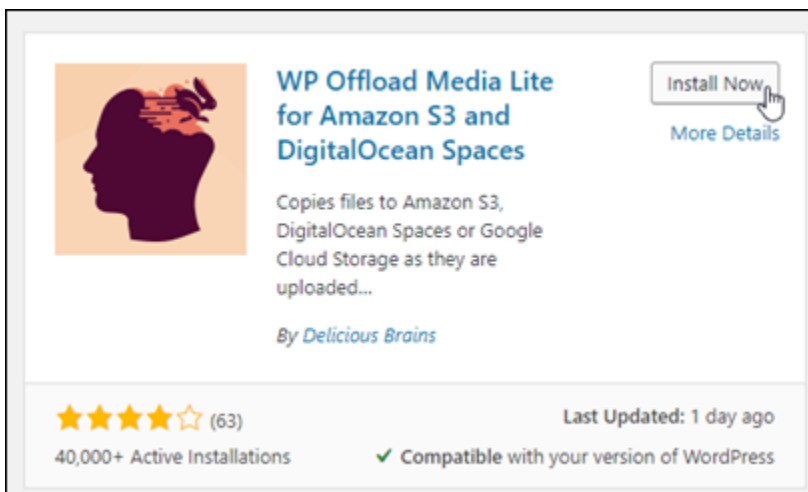
1. Sign in to the dashboard of your WordPress website as an administrator.

For more information, see [Getting the application user name and password for your Bitnami instance in Amazon Lightsail](#).

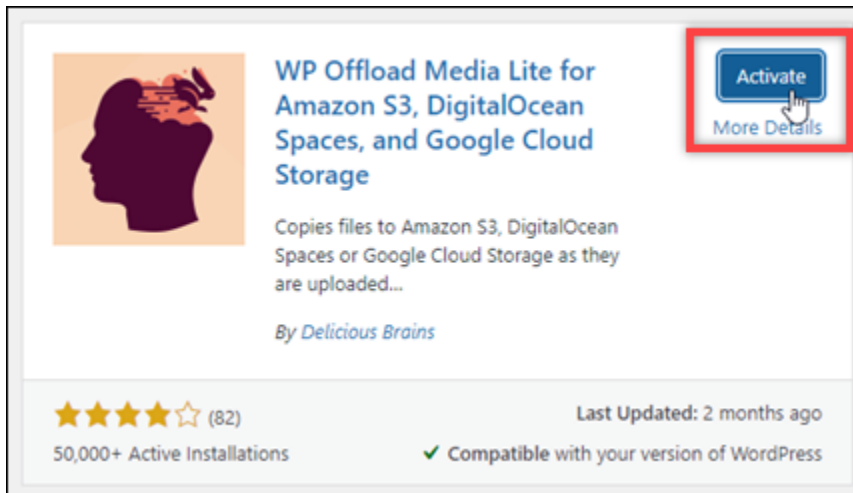
2. Pause on **Plugins** in the left navigation menu, and choose **Add New**.



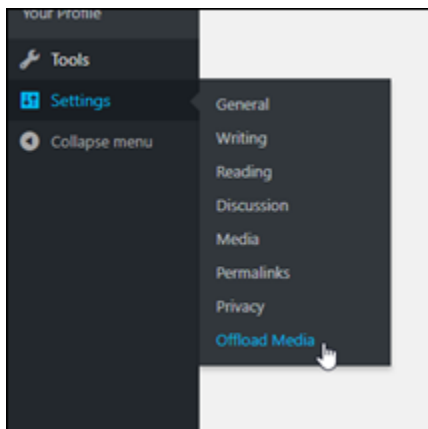
3. Search for **WP Offload Media Lite**.
4. In the search results, choose **Install Now** next to the **WP Offload Media Lite** plugin.



5. Choose **Activate** after the plugin is done installing.




6. In the left navigation menu, choose **Settings**, then choose **Offload Media**.



7. In the **Offload Media Lite** page, choose **Amazon S3** as the storage provider.

Offload Media Lite Media Library Addons Support


STORAGE PROVIDER


 **Amazon S3**

Define access keys in wp-config.php

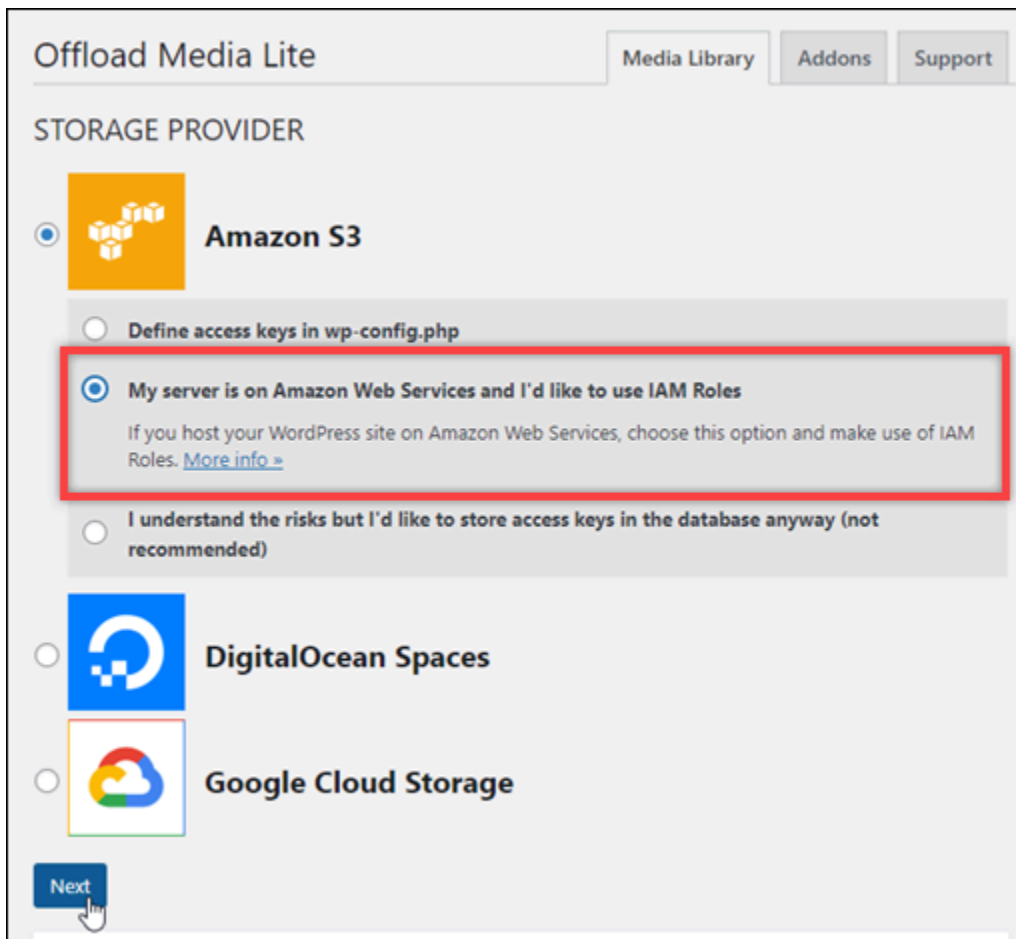
My server is on Amazon Web Services and I'd like to use IAM Roles
If you host your WordPress site on Amazon Web Services, choose this option and make use of IAM Roles. [More info >](#)

I understand the risks but I'd like to store access keys in the database anyway (not recommended)

 **DigitalOcean Spaces**

 **Google Cloud Storage**

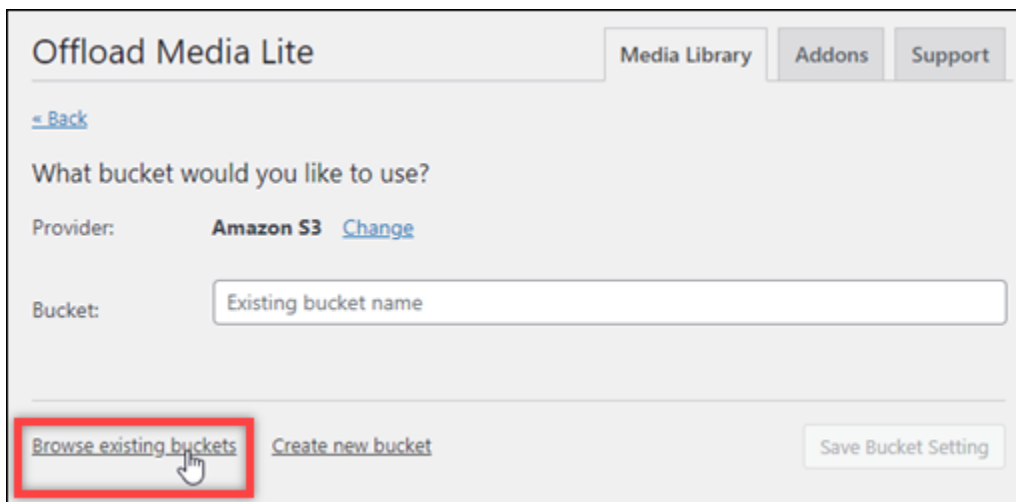
8. Choose **My server is on Amazon Web Services and I'd like to use IAM Roles**.



The screenshot shows the 'Offload Media Lite' configuration interface. At the top, there are navigation tabs for 'Media Library', 'Addons', and 'Support'. Below this is the 'STORAGE PROVIDER' section. Three options are listed: 'Amazon S3', 'DigitalOcean Spaces', and 'Google Cloud Storage'. The 'Amazon S3' option is selected, and its sub-options are visible. A red box highlights the sub-option 'My server is on Amazon Web Services and I'd like to use IAM Roles', which is also selected. Below this, there is a 'Next' button with a mouse cursor pointing to it.

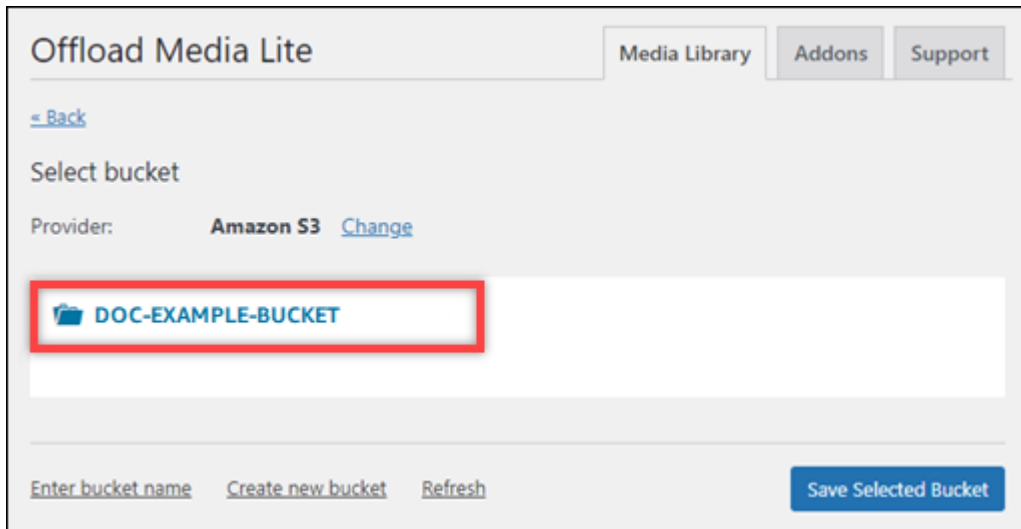
9. Choose **Next**.

10. Choose **Browse existing buckets** in the **What bucket would you like to use?** page that appears.



The screenshot shows the 'What bucket would you like to use?' configuration page. At the top, there are navigation tabs for 'Media Library', 'Addons', and 'Support'. Below this is a '- Back' link. The main heading is 'What bucket would you like to use?'. Underneath, the 'Provider' is set to 'Amazon S3' with a 'Change' link. The 'Bucket' field contains the text 'Existing bucket name'. At the bottom, there are three buttons: 'Browse existing buckets' (highlighted with a red box and a mouse cursor), 'Create new bucket', and 'Save Bucket Setting'.

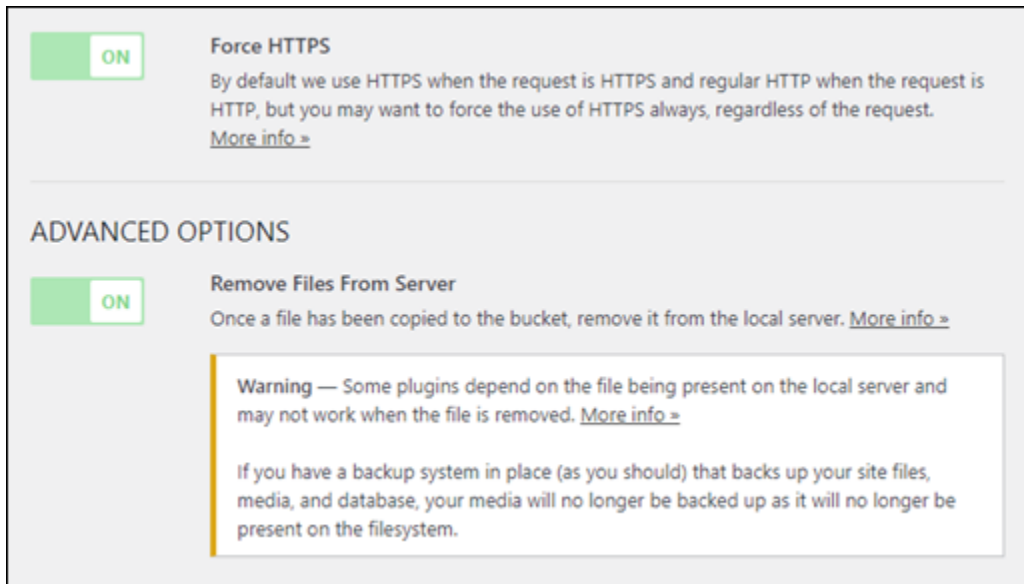
11. Choose the name of the bucket that you created to use with your WordPress instance.



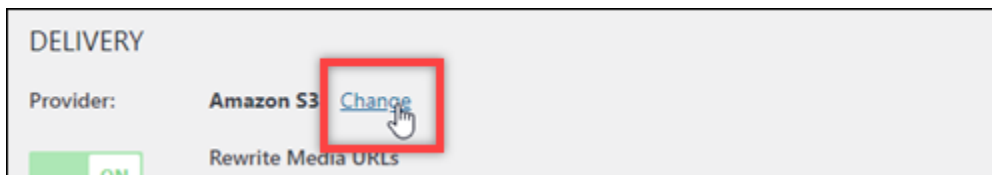
12. In the **Offload Media Lite Settings** page that appears, turn on **Force HTTPS** and **Remove Files From Server**.

- The **Force HTTPS** setting must be turned on because Lightsail buckets use HTTPS by default to serve media files. If you don't turn this feature on, media files that are uploaded to your Lightsail bucket from your WordPress website won't be served correctly to your website visitors.

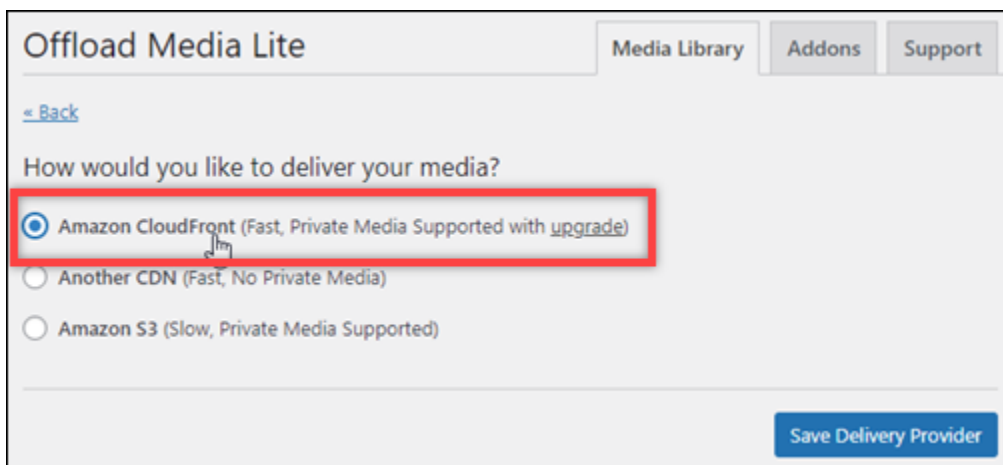
The **Remove Files From Server** setting ensures that media that is uploaded to your Lightsail bucket isn't also stored on your instance's disk. If you don't turn this feature on, media files that are uploaded to your Lightsail bucket are also stored on the local storage of your WordPress instance.



13. Under the **Delivery** section of the page, choose **Change** next to the Amazon S3 label.



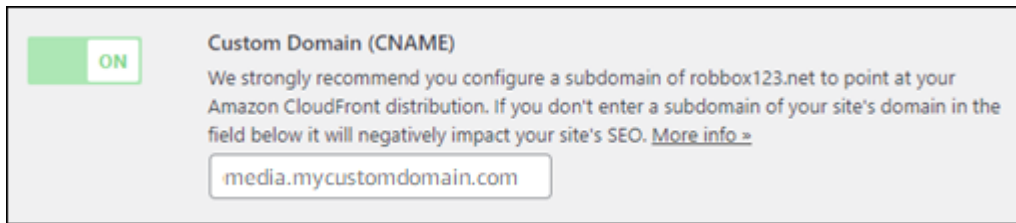
14. In the **How would you like to deliver your media?** page that appears, select **Amazon CloudFront**.



15. Choose **Save Delivery Provider**.

16. In the **Offload Media Lite Settings** page that appears, turn on **Custom Domain (CNAME)**. Then, enter the domain of your Lightsail distribution into the text box. This could be the

default domain of your distribution (for example, `123abc.cloudfront.net`) or the custom domain for your distribution (for example, `media.mycustomdomain.com`), if you enabled it.



17. Choose **Save Changes**.

Note

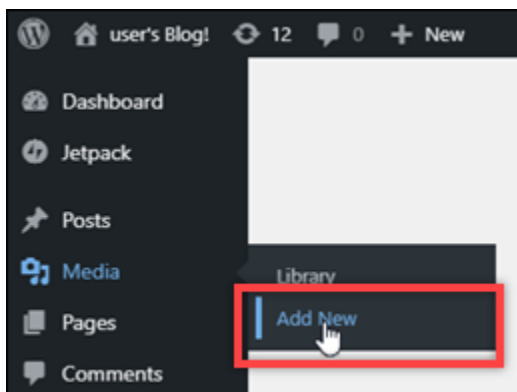
To return to the **Offload Media Lite Settings** page later, pause on **Settings** in the left navigation menu, and choose **Offload Media**.

Your WordPress website is now configured to use the Media Lite Plugin. The next time you upload a media file through WordPress, that file is automatically uploaded to your Lightsail bucket, and is served by the distribution. To test the configuration, continue to the next section of this tutorial.

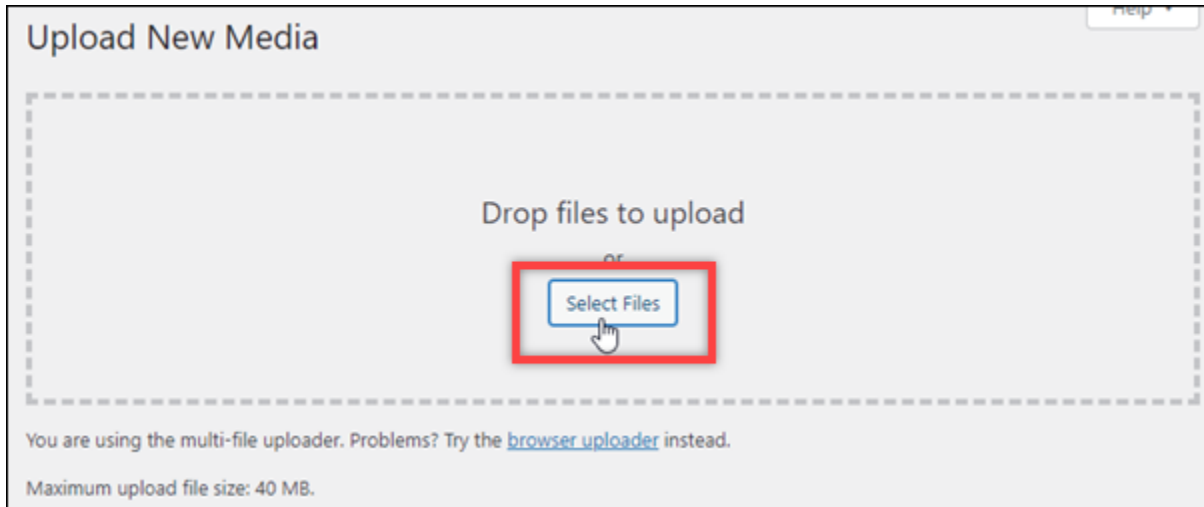
Step 6: Test the connection between your WordPress website and your Lightsail bucket and distribution

Complete the following procedure to upload a media file to your WordPress instance and confirm that it is uploaded to your Lightsail bucket and is served from your distribution.

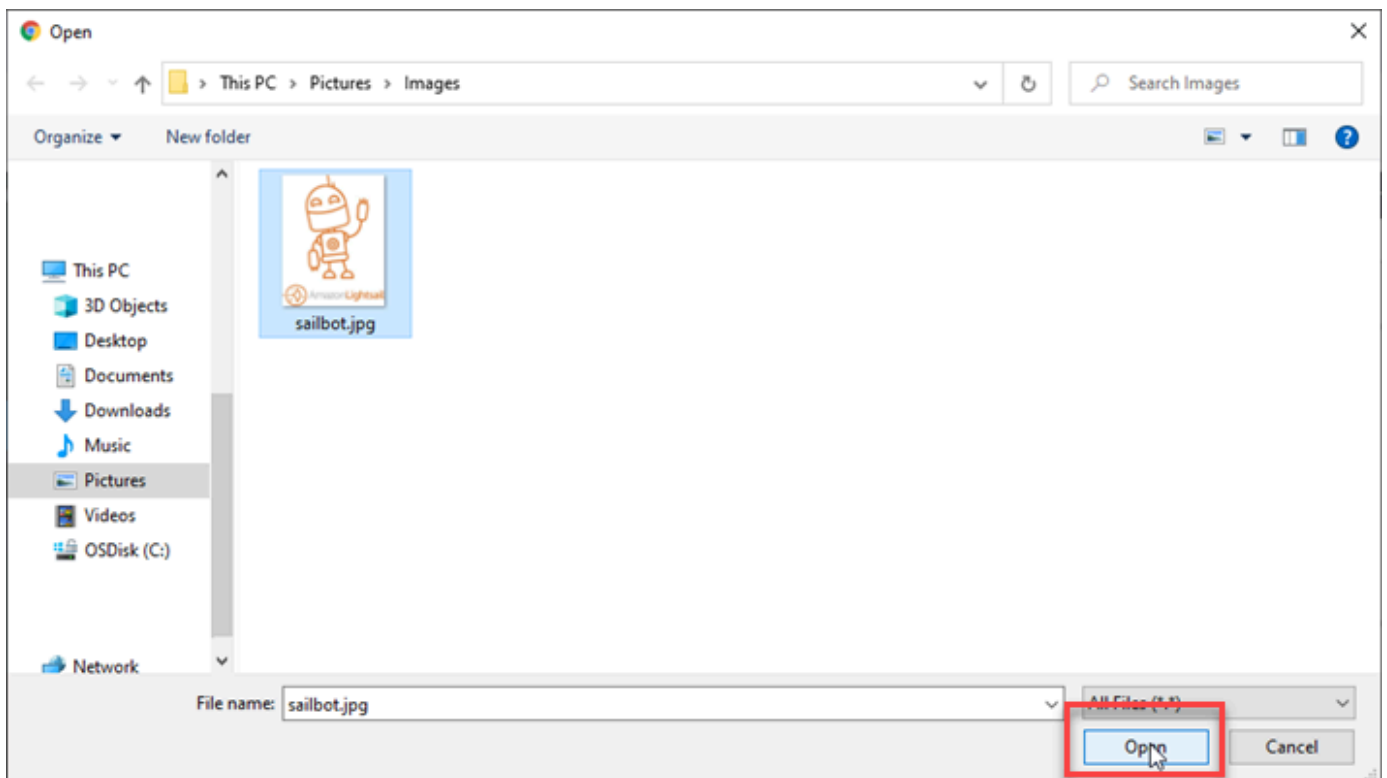
1. Pause on **Media** in the left navigation menu of the WordPress dashboard, and choose **Add New**.



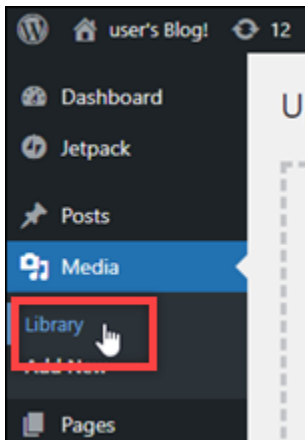
2. Choose **Select Files** on the **Upload New Media** page that appears.



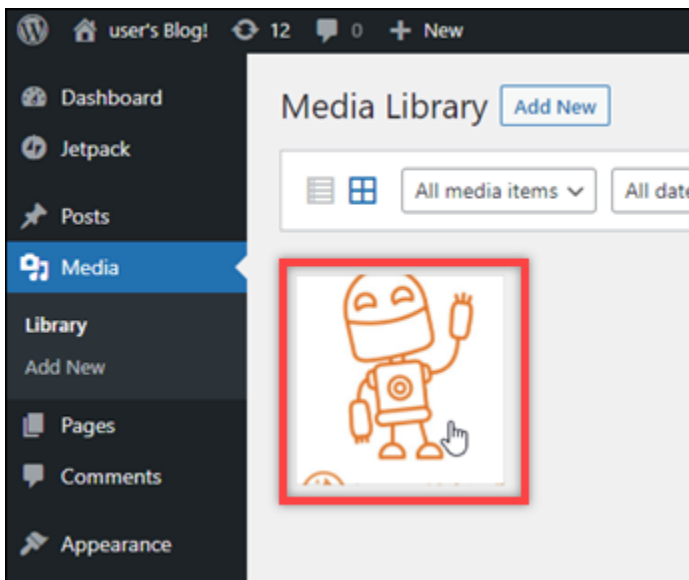
3. Choose a media file to upload from your local computer, and choose **Open**.



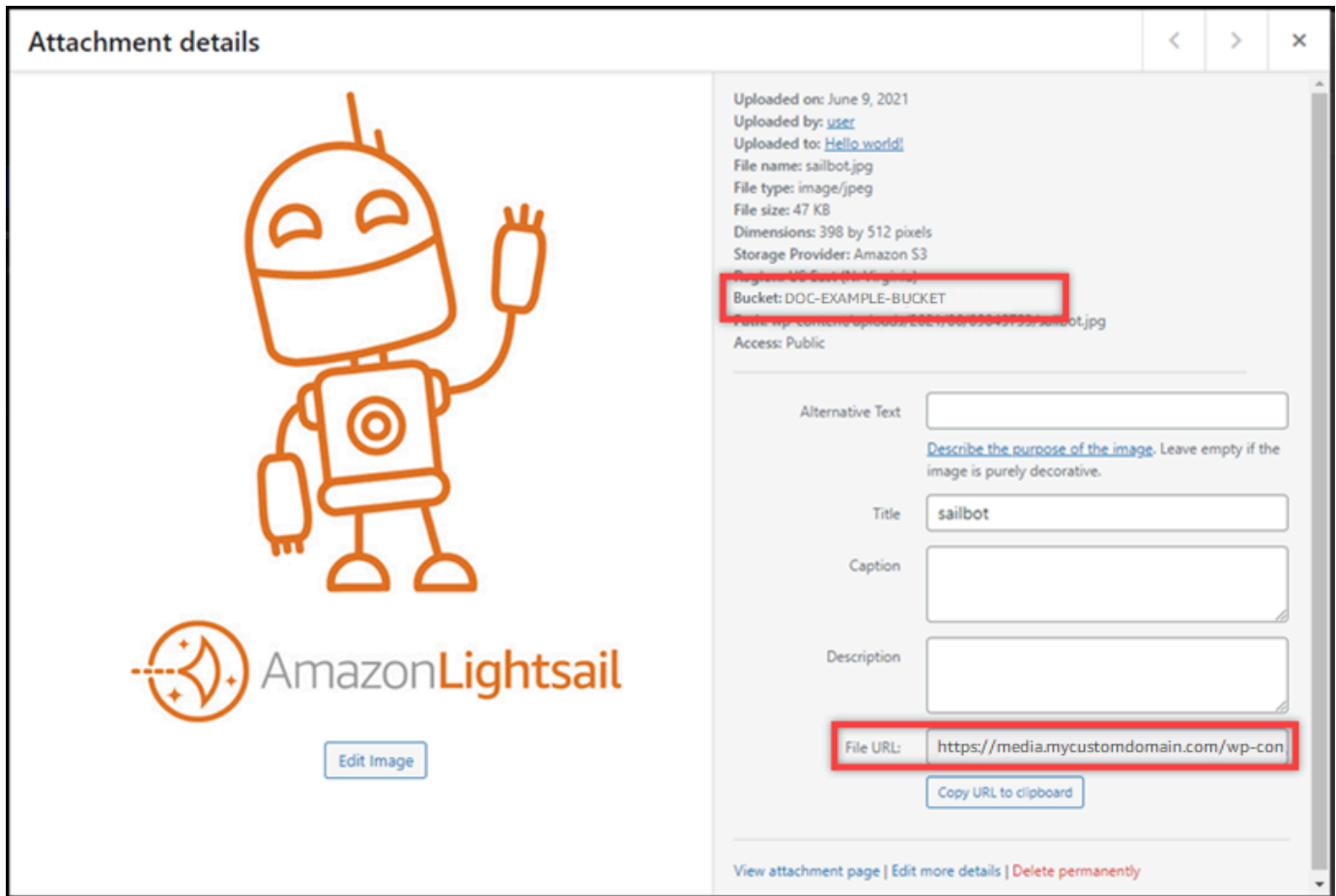
4. When the file is done uploading, choose **Library** under **Media** in the left navigation menu.



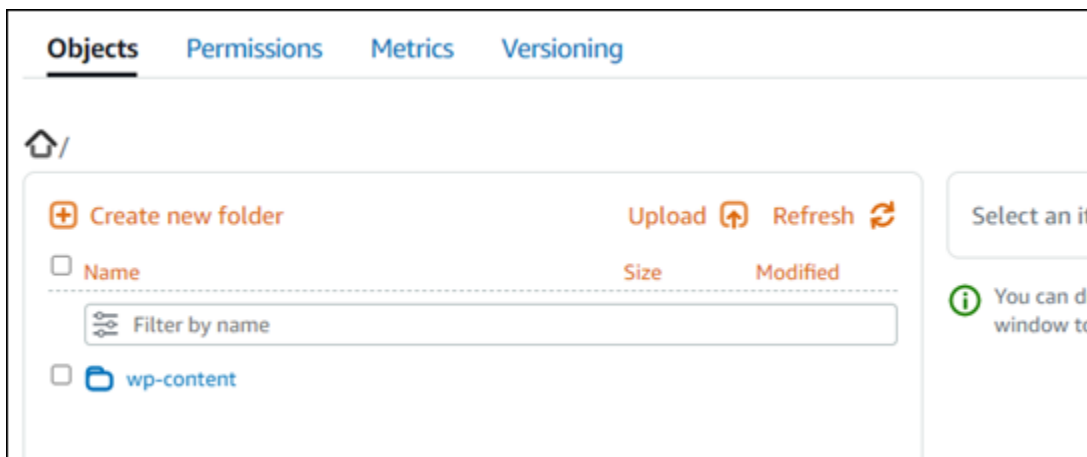
5. Choose the file that you recently uploaded.



6. In the details panel of the file, the name of your bucket appears in the **Bucket** field. The URL of your distribution appears in the **File URL** field.



- If you go to the **Objects** tab of the Lightsail bucket management page, you should see a **wp-content** folder. This folder is created by the Offload Media Lite plugin, and is used to store your uploaded media files.



Manage buckets and objects

These are the general steps to manage your Lightsail object storage bucket:

1. Learn about objects and buckets in the Amazon Lightsail object storage service. For more information, see [Object storage in Amazon Lightsail](#).
2. Learn about the names that you can give your buckets in Amazon Lightsail. For more information, see [Bucket naming rules in Amazon Lightsail](#).
3. Get started with the Lightsail object storage service by creating a bucket. For more information, see [Creating buckets in Amazon Lightsail](#).
4. Learn about security best practices for buckets and the access permissions that you can configure for your bucket. You can make all objects in your bucket public or private, or you can choose to make individual objects public. You can also grant access to your bucket by creating access keys, attaching instances to your bucket, and granting access to other AWS accounts. For more information, see [Security Best Practices for Amazon Lightsail object storage](#) and [Understanding bucket permissions in Amazon Lightsail](#).

After learning about bucket access permissions, see the following guides to grant access to your bucket:

- [Block public access for buckets in Amazon Lightsail](#)
 - [Configuring bucket access permissions in Amazon Lightsail](#)
 - [Configuring access permissions for individual objects in a bucket in Amazon Lightsail](#)
 - [Creating access keys for a bucket in Amazon Lightsail](#)
 - [Configuring resource access for a bucket in Amazon Lightsail](#)
 - [Configuring cross-account access for a bucket in Amazon Lightsail](#)
5. Learn how to enable access logging for your bucket, and how to use access logs to audit the security of your bucket. For more information, see the following guides.
 - [Access logging for buckets in the Amazon Lightsail object storage service](#)
 - [Access log format for a bucket in the Amazon Lightsail object storage service](#)
 - [Enabling access logging for a bucket in the Amazon Lightsail object storage service](#)
 - [Using access logs for a bucket in Amazon Lightsail to identify requests](#)
 6. Create an IAM policy that grants a user the ability to manage a bucket in Lightsail. For more information, see [IAM policy to manage buckets in Amazon Lightsail](#).

7. Learn about the way that objects in your bucket are labeled and identified. For more information, see [Understanding object key names in Amazon Lightsail](#).
8. Learn how to upload files and manage objects in your buckets. For more information, see the following guides.
 - [Uploading files to a bucket in Amazon Lightsail](#)
 - [Uploading files to a bucket in Amazon Lightsail using multipart upload](#)
 - [Viewing objects in a bucket in Amazon Lightsail](#)
 - [Copying or moving objects in a bucket in Amazon Lightsail](#)
 - [Downloading objects from a bucket in Amazon Lightsail](#)
 - [Filtering objects in a bucket in Amazon Lightsail](#)
 - [Tagging objects in a bucket in Amazon Lightsail](#)
 - [Deleting objects in a bucket in Amazon Lightsail](#)
9. Enable object versioning to preserve, retrieve, and restore every version of every object stored in your bucket. For more information, see [Enabling and suspending object versioning in a bucket in Amazon Lightsail](#).
10. After enabling object versioning, you can restore previous versions of objects in your bucket. For more information, see [Restoring previous versions of objects in a bucket in Amazon Lightsail](#).
11. Monitor the utilization of your bucket. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#).
12. Configure an alarm for bucket metrics to be notified when the utilization of your bucket crosses a threshold. For more information, see [Creating bucket metric alarms in Amazon Lightsail](#).
13. Change the storage plan of your bucket if it's running low on storage and network transfer. For more information, see [Changing the plan of your bucket in Amazon Lightsail](#).
14. Learn how to connect your bucket to other resources. For more information, see the following tutorials.
 - [Tutorial: Connecting a WordPress instance to an Amazon Lightsail bucket](#)
 - [Tutorial: Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#)
15. Delete your bucket if you're no longer using it. For more information, see [Deleting buckets in Amazon Lightsail](#).

Use Lightsail with other AWS services

Amazon Lightsail uses a focused set of AWS services like Amazon EC2 and AWS Identity and Access Management to make it easier to get started. But that doesn't mean you're limited to those services!

You can integrate Lightsail resources with other AWS services through Amazon VPC peering. [Learn how to set up VPC peering.](#)

Follow the links below to learn more about other AWS services.

Virtual machines (virtual private servers)

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It's designed to make web-scale cloud computing easier for developers.

With Amazon EC2 you can obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, so you can quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by enabling you to pay only for capacity that you actually use. Amazon EC2 provides developers with tools to build failure resilient applications and isolate themselves from common failure scenarios.

[Learn more about Amazon EC2.](#)

Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud, where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

You can easily customize the network configuration for your Amazon VPC. For example, you can create a public-facing subnet for your web servers that has access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access. You can leverage multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

Additionally, you can create a Hardware Virtual Private Network (VPN) connection between your corporate datacenter and your VPC and leverage the AWS Cloud as an extension of your corporate datacenter.

[Learn more about Amazon VPC.](#)

Serverless computing

AWS Lambda

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code isn't running. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

[Learn more about AWS Lambda.](#)

Amazon API Gateway

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create an API that acts as a "front door" for applications to access data, business logic, or functionality from your backend services. These include workloads running on Amazon EC2, code running on Lambda, or any Web application. Amazon API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls. These include traffic management, authorization and access control, monitoring, and API version management. Amazon API Gateway has no minimum fees or startup costs. You pay only for the API calls you receive and the amount of data transferred out.

[Learn more about Amazon API Gateway.](#)

Databases

Amazon DynamoDB

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It's a fully managed cloud

database and supports both document and key-value store models. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad tech, IoT, and many other applications.

[Learn more about DynamoDB.](#)

Amazon RDS

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you to focus on your applications and business. Amazon RDS provides you six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server.

[Learn more about Amazon RDS.](#)

Amazon Aurora

Amazon Aurora is a MySQL-compatible relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases. Aurora provides up to five times better performance than MySQL with the security, availability, and reliability of a commercial database at one tenth the cost.

[Learn more about Amazon Aurora.](#)

Load balancers

Elastic Load Balancing

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications, seamlessly providing the required amount of load balancing capacity needed to route application traffic.

Elastic Load Balancing offers two types of load balancers. Both feature high availability, automatic scaling, and robust security. These include the Classic Load Balancer that routes traffic based on either application or network-level information, and the Application Load Balancer that routes traffic based on advanced application-level information that includes the content of the request. The Classic Load Balancer is ideal for simple load balancing of traffic across multiple Amazon EC2 instances. The Application Load Balancer is ideal for applications needing advanced routing capabilities, microservices, and container-based architectures.

Application Load Balancer offers the ability to route traffic to multiple services or to load balance across multiple ports on the same Amazon EC2 instance.

[Learn more about Elastic Load Balancing.](#)

Application Load Balancer

An Application Load Balancer is a load balancing option for the Elastic Load Balancing service that operates at the application layer and allows you to define routing rules based on content across multiple services or containers running on one or more Amazon EC2 instances.

[Learn more about Application Load Balancer.](#)

Big data

Amazon Kinesis services

Amazon Kinesis services make it easy to work with real-time streaming data in the AWS cloud. Amazon Kinesis services include the following: [Amazon Data Firehose](#) to easily load massive volumes of streaming data into AWS, [Amazon Managed Service for Apache Flink](#) to analyze streaming data with standard SQL, and [Amazon Kinesis Data Streams](#) to build your own custom applications that process or analyze streaming data.

[Learn more about Amazon Kinesis services.](#)

Amazon EMR

Amazon EMR provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances. You can also run other popular distributed frameworks such as Apache Spark, HBase, Presto, and Flink in Amazon EMR, and interact with data in other AWS data stores such as Amazon S3 and DynamoDB.

Amazon EMR securely and reliably handles a broad set of big data use cases, including log analysis, web indexing, data transformations (ETL), machine learning, financial analysis, scientific simulation, and bioinformatics.

[Learn more about Amazon EMR.](#)

Amazon Redshift

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse that makes it simple and cost-effective to analyze all your data using your existing business intelligence tools.

[Learn more about Amazon Redshift.](#)

Storage

Amazon Simple Storage Service (Amazon S3)

Amazon S3, provides developers and IT teams with secure, durable, highly scalable cloud storage. Amazon S3 is easy-to-use object storage, with a simple web service interface to store and retrieve any amount of data from anywhere on the web. With Amazon S3, you pay only for the storage you actually use. There's no minimum fee and no setup cost.

Amazon S3 offers a range of storage classes designed for different use cases including Amazon S3 Standard for general-purpose storage of frequently accessed data, Amazon S3 Standard - Infrequent Access (Standard - IA) for long-lived, but less frequently accessed data, and S3 Glacier for long-term archive. Amazon S3 also offers configurable lifecycle policies for managing your data throughout its lifecycle. Once a policy is set, your data automatically migrates to the most appropriate storage class without any changes to your applications.

Amazon S3 can be used alone or together with other AWS services such as Amazon EC2 and IAM, as well as cloud data migration services and gateways for initial or ongoing data ingestion. Amazon S3 provides cost-effective object storage for a wide variety of use cases including backup and recovery, nearline archive, big data analytics, disaster recovery, cloud applications, and content distribution.

[Learn more about Amazon S3.](#)

Amazon Elastic Block Store (Amazon EBS)

Amazon EBS provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. Amazon EBS volumes offer the consistent and low-latency performance needed to run your workloads. With Amazon EBS, you can scale your usage up or down within minutes – all while paying a low price for only what you provision.

[Learn more about Amazon EBS.](#)

Monitoring and alarms

Amazon CloudWatch

Amazon CloudWatch is a monitoring service for AWS Cloud resources and the applications you run on AWS. You can use CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react and keep your application running smoothly.

[Learn more about Amazon CloudWatch.](#)

Application deployment

AWS Elastic Beanstalk

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

[Learn more about Elastic Beanstalk.](#)

Application containers

Amazon Elastic Container Service (Amazon ECS)

Amazon ECS is a highly scalable, high-performance container management service that supports Docker containers and enables you to easily run applications on a managed cluster of Amazon EC2 instances. Amazon ECS eliminates the need for you to install, operate, and scale your own cluster management infrastructure. With simple API calls, you can launch and stop Docker-enabled applications, query the complete state of your cluster, and access many familiar

features like security groups, Elastic Load Balancing, Amazon EBS volumes, and IAM roles. You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs and availability requirements. You can also integrate your own scheduler or third-party schedulers to meet business or application-specific requirements.

[Learn more about Amazon ECS.](#)

Security and User Sign-in

AWS Identity and Access Management (IAM)

IAM lets you securely control access to AWS services and resources for your users. Using IAM, you can create and manage AWS users and groups and use permissions to allow and deny their access to AWS resources.

[Learn more about IAM.](#)

Amazon Cognito User Pools

Amazon Cognito lets you easily add user sign-up and sign-in to your mobile and web apps. With Amazon Cognito, you also have the options to authenticate users through social identity providers such as Facebook, Twitter, or Amazon, with SAML identity solutions, or by using your own identity system. In addition, Amazon Cognito enables you to save data locally on users' devices, allowing your applications to work even when the devices are offline. You can then synchronize data across users' devices so that their app experience remains consistent, regardless of the device they use.

With Amazon Cognito, you can focus on creating great app experiences instead of worrying about building, securing, and scaling a solution to handle user management, authentication, and sync across devices.

[Learn more about Amazon Cognito.](#)

Source Control and Application Lifecycle Management

AWS CodeCommit

AWS CodeCommit is a fully managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories. AWS CodeCommit eliminates the

need to operate your own source control system or worry about scaling its infrastructure. You can use AWS CodeCommit to securely store anything from source code to binaries, and it works seamlessly with your existing Git tools.

[Learn more about AWS CodeCommit.](#)

Queues and Messaging

Amazon SQS

Amazon Simple Queue Service (Amazon SQS) is a fast, reliable, scalable, fully managed message queuing service. Amazon SQS makes it simple and cost-effective to decouple the components of a cloud application. You can use Amazon SQS to transmit any volume of data, without losing messages or requiring other services to be always available. Amazon SQS includes *standard queues* with high throughput and at-least-once processing, and *FIFO queues* that provide FIFO (first-in, first-out) delivery and exactly-once processing.

With Amazon SQS, you can offload the administrative burden of operating and scaling a highly available messaging cluster, while paying a low price for only what you use.

[Learn more about Amazon SQS.](#)

Amazon SNS

Amazon Simple Notification Service (Amazon SNS) is a fast, flexible, fully managed push notification service that lets you send individual messages or to fan out messages to large numbers of recipients. Amazon SNS makes it simple and cost-effective to send push notifications to mobile device users or email recipients, or even to send messages to other distributed services.

With Amazon SNS, you can send notifications to Apple Push Notification Service (APNS), Google Cloud Messaging (GCM), Fire OS, and Windows devices, as well as to Android devices in China with Baidu Cloud Push. You can use Amazon SNS to send SMS messages to mobile device users worldwide.

Beyond these endpoints, Amazon SNS can also deliver messages to Amazon SQS, AWS Lambda functions, or to any HTTP endpoint.

[Learn more about Amazon SNS.](#)

Amazon SES

Amazon Simple Email Service (Amazon SES) is a cost-effective email service built on the reliable and scalable infrastructure that Amazon.com developed to serve its own customer base. With Amazon SES, you can send and receive email with no required minimum commitments. You pay as you go, and you only pay for what you use.

[Learn more about Amazon SES.](#)

Workflow

Amazon Simple Workflow Service (Amazon SWF)

Amazon SWF helps developers build, run, and scale background jobs that have parallel or sequential steps. You can think of Amazon SWF as a fully managed state tracker and task coordinator in the cloud.

If your app's steps take more than 500 milliseconds to complete, you need to track the state of processing, and you need to recover or retry if a task fails. Amazon SWF can help you.

[Learn more about Amazon SWF.](#)

Streaming applications

Amazon AppStream

Amazon AppStream lets you deliver your Windows applications to any device.

Amazon AppStream enables you to stream your existing Windows applications from the cloud, reaching more users on more devices, without code modifications. With Amazon AppStream, your application is deployed and rendered on AWS infrastructure and the output is streamed to mass-market devices, such as personal computers, tablets, and mobile phones. Because your application is running in the cloud, it can scale to handle vast computational and storage needs, regardless of the devices your customers are using. Amazon AppStream provides an SDK for streaming your application from the cloud. You can integrate your own custom clients, subscriptions, identity, and storage solution with Amazon AppStream to build a custom streaming solution that meets the needs of your business.

[Learn more about Amazon AppStream.](#)

Create Lightsail resources with AWS CloudFormation

Amazon Lightsail is integrated with AWS CloudFormation, a service that helps you to model and set up your AWS resources so that you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want (such as instances and disks), and AWS CloudFormation provisions and configures those resources for you.

When you use AWS CloudFormation, you can reuse your template to set up your Lightsail resources consistently and repeatedly. Describe your resources once, and then provision the same resources over and over in multiple AWS accounts and Regions.

Lightsail and AWS CloudFormation templates

To provision and configure resources for Lightsail and related services, you must understand [AWS CloudFormation templates](#). Templates are formatted text files in JSON or YAML. These templates describe the resources that you want to provision in your AWS CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use AWS CloudFormation Designer to help you get started with AWS CloudFormation templates. For more information, see [What is AWS CloudFormation Designer?](#) in the *AWS CloudFormation User Guide*.

Lightsail supports creating instances and disks in AWS AWS CloudFormation. For more information, see the [Lightsail resource type reference](#) in the *AWS CloudFormation User Guide*.

Learn more about AWS CloudFormation

To learn more about AWS CloudFormation, see the following resources:

- [AWS CloudFormation](#)
- [AWS CloudFormation User Guide](#)
- [AWS CloudFormation API Reference](#)
- [AWS CloudFormation Command Line Interface User Guide](#)

AWS CloudFormation stacks for Lightsail

Amazon Lightsail uses AWS CloudFormation to create Amazon Elastic Compute Cloud (Amazon EC2) instances from exported snapshots. A CloudFormation stack is created when you request to

create an Amazon EC2 instance using the Lightsail console or Lightsail API. The stack performs a series of actions in your Amazon Web Services (AWS) account to create all of the related resources for the instance, such as the Amazon EC2 instance from an Amazon Machine Image (AMI), the Elastic Block Store (EBS) system volume from an EBS snapshot, and the security group for the instance. To learn more about AWS CloudFormation stacks, see [Working with Stacks](#) in the AWS CloudFormation documentation.

You can access the AWS CloudFormation stacks through the Lightsail console or in the AWS CloudFormation console. This guide shows you how to access both.

Note

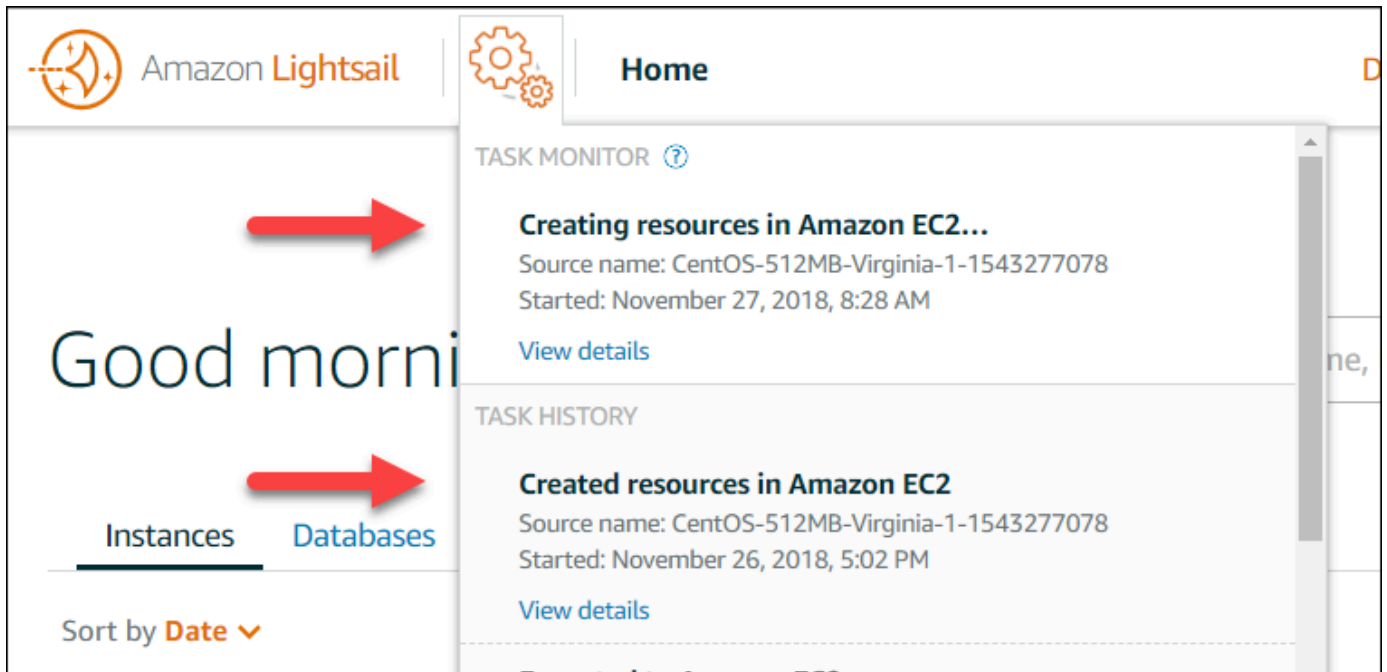
The AWS CloudFormation stack used to create your Amazon EC2 resources is permanently linked to your Amazon EC2 resources. If you delete the stack, then all related resources are automatically deleted. Because of this, you should not delete any of the AWS CloudFormation stacks created by Lightsail, and instead delete your Amazon EC2 resources using the EC2 console.

Accessing the AWS CloudFormation stacks through the Lightsail console

After you choose to create an instance in Amazon EC2 using the Lightsail console or the Lightsail API, an AWS CloudFormation stack is created and its status is tracked using the task monitor. To learn more about the task monitor, see [Task monitor](#).

To view your AWS CloudFormation stacks in the Lightsail console

1. Sign in to the [Lightsail console](#).
2. Choose the task monitor in the top navigation pane.
3. To access a CloudFormation stack for a previously created Amazon EC2 instance, choose **View details** for a task labeled with **Creating resources in Amazon EC2** or **Created resources in Amazon EC2**.



4. The confirmation page that appears lists the CloudFormation stack for the task. Choose the stack name to open the stack details in the AWS CloudFormation console.

Accessing the stacks in the AWS CloudFormation console

You can also access your stack details through the [AWS CloudFormation console](#). The stacks created by Lightsail begin with “Lightsail-stack” and have a description of “CloudFormation stack used to create Amazon EC2 resources” as shown in the following screenshot.

Stacks with a **CREATE_IN_PROGRESS** status are in the process of creating Amazon EC2 resources from your exported Lightsail snapshots. Stacks with a **CREATE_COMPLETED** status have completed the process of creating Amazon EC2 resources. To view the resources created by a stack, choose the checkbox next to the stack name, and then choose the **Resources** tab.

Create Stack ▾
Actions ▾
Design template
↻ ⚙

Filter: Active ▾

Showing 4 stacks

	Stack Name	Created Time	Status	Drift Status	Description
<input checked="" type="checkbox"/>	Lightsail-Stack-a0e00482-77a3-4f32-a3...	2018-11-19 09:46:24 UTC-0800	CREATE_COMPLETE	NOT_CHECKED	CloudFormation stack used...
<input type="checkbox"/>	Lightsail-Stack-104e982e-cba3-49d7-96...	2018-11-19 09:15:51 UTC-0800	CREATE_COMPLETE	NOT_CHECKED	CloudFormation stack used...
<input type="checkbox"/>	Lightsail-Stack-f4267e8-44c6-49e0-941...	2018-11-12 11:17:42 UTC-0800	CREATE_COMPLETE	NOT_CHECKED	CloudFormation stack used...
<input type="checkbox"/>	Lightsail-Stack-0e805e88-f78a-4c4e-85...	2018-11-02 14:35:24 UTC-0700	CREATE_COMPLETE	NOT_CHECKED	CloudFormation stack used...

Overview
Outputs
Resources
Events
Template
Parameters
Tags
Stack Policy
Change Sets
Rollback Triggers

☰ ☰ ☰

To view detailed drift information for specific resources, visit the [Drift Details page](#). ⓘ

Logical ID	Physical ID	Type	Drift Status	Status	Status Reason
Instance3fd67c5c...	i-09a6442334a538516	AWS::EC2::Instance	NOT_CHECKED	CREATE_COMPL...	
SecurityGroup9e8...	sg-0359d91e0b64c4556	AWS::EC2::SecurityGroup	NOT_CHECKED	CREATE_COMPL...	

Amazon Lightsail billing

Billing for Amazon Lightsail is handled through Amazon Web Services (AWS) billing. To view your Lightsail bill, go to the [AWS Billing and Cost Management Dashboard](#), or choose **Billing** on the top navigation bar of the Lightsail console. For more information about pricing, see the [Lightsail pricing page](#).

View your detailed Lightsail bill

To view a detailed breakdown of your monthly Lightsail bill:

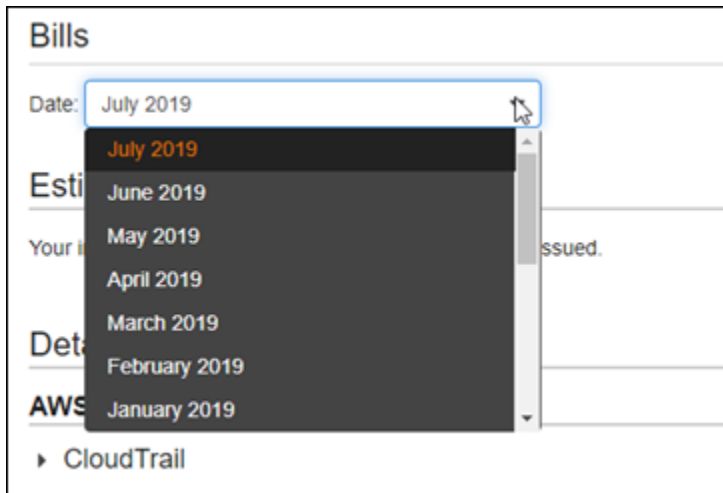
1. Sign in to the [AWS Billing and Cost Management Dashboard](#).

The billing dashboard home page displays a high-level month-to-date breakdown of your bill.

2. Choose **Bill Details** on the dashboard home page, or choose **Bills** in the left navigation pane, to view a detailed version of your monthly bill.

Service	Amount
Lightsail	\$196.53
EC2	\$0.91
Route53	\$0.50
GuardDuty	\$0.26

3. Choose the **Date** drop-down menu to select a month other than the current month.



4. Scroll down on the **Bills** page, and expand the Lightsail line item to view detailed usage for each region.

▼ Lightsail		\$192.69
▶ US East (N. Virginia)		\$0.00
▼ US West (Oregon)		\$192.69
Amazon Lightsail Bundle:0.5GB		\$6.22
\$0.0047 / Hour of 0.5GB bundle Instance	1,323.603 Hrs	\$6.22
Amazon Lightsail Bundle:1GB		\$0.16
\$0.00672/ Hour of 1GB bundle Instance	23.073 Hrs	\$0.16
Amazon Lightsail Bundle:4GB		\$19.35
\$0.0269 / Hour of 4GB bundle Instance	720 Hrs	\$19.35
Amazon Lightsail Bundle:8GB		\$116.12
\$0.0538 / Hour of 8GB bundle Instance	2,160 Hrs	\$116.12

Billing usage types

The following list describes the usage types that appear in your Lightsail billing and usage reports. These usage types help identify the charges on your monthly bill for Lightsail resources.

Note

For the following usage types that specify a **Region** code, see the [Region codes in your bill](#) section of this guide to identify the corresponding AWS Region.

- **Amazon Lightsail Bundle:SizeGB:** The Linux or Unix instance plan used (in hours). The **Size** defines the memory specification of the instance plan used. For example, if **4GB** of memory is specified, then the billed hours for the \$20 USD/month Linux or Unix instance plan is displayed.
- **Amazon Lightsail Bundle:SizeGB (Windows):** The Windows instance plan used (in hours). The **Size** defines the memory specification of the instance plan used. For example, if **4GB** of memory is specified, then the billed hours for the \$40 USD/month Windows instance plan is displayed.
- **Amazon Lightsail RelationalDatabase:SizeGB:** The standard database plans used (in hours). The **Size** defines the memory specification of the database plan used. For example, if **4GB** of memory is specified, then the billed hours for the \$60 USD/month standard database plan is displayed.
- **Amazon Lightsail RelationalDatabase:SizeGB (high availability):** The high availability database plans used (in hours). The **Size** defines the memory specification of the database plan used. For example, if **4GB** of memory is specified, then the billed hours for the \$120 USD/month high availability database plan is displayed.
- **Amazon Lightsail Region-DiskUsage:** The amount of block storage disk used (in gigabytes per month).
- **Amazon Lightsail DNS-Queries:** The number (count) of DNS queries for the month.
- **Amazon Lightsail Load Balancer:** The amount of load balancers used (in hours).
- **Amazon Lightsail Region-SnapshotUsage:** The amount of stored snapshot data (in gigabytes per month).
- **Amazon Lightsail Region-UnusedStaticIP:** The amount of un-attached static IPs (in hours).
- **Amazon Lightsail Region-TotalDataXfer-In-Bytes:** The total amount of data transferred in (in gigabytes).
- **Amazon Lightsail Region-TotalDataXfer-Out-Bytes:** The total amount of data transferred out (in gigabytes).
- **Amazon Lightsail Region-DataXfer-Out-Overage-Bytes:** The amount of data transferred out to the internet or public IPs that is over the allowance of the instance or database plan(s) used (in gigabytes).
- **Amazon Lightsail Region-DataXfer-Out-Free-Bytes (deprecated):** The amount of data transferred out that is within the allowance of the instance or database plan(s) used (in gigabytes).
- **Amazon Lightsail Region-DataXfer-Out-Other-Bytes (deprecated):** The amount of data transferred out to private IPs that is over the allowance of the instance or database plan(s) used (in gigabytes). This overage is free when the transfer is to an AWS resource over a private IP.

Region codes in your bill

Lightsail billing and usage reports use codes and abbreviations. For example, for usage type, region is replaced with one of the following abbreviations:

- **APN1:** Asia Pacific (Tokyo) (ap-northeast-1)
- **APN2:** Asia Pacific (Seoul) (ap-northeast-2)
- **APS1:** Asia Pacific (Singapore) (ap-southeast-1)
- **APS2:** Asia Pacific (Sydney) (ap-southeast-2)
- **APS3:** Asia Pacific (Mumbai) (ap-south-1)
- **CAN1:** Canada (Central) (ca-central-1)
- **EU:** EU (Ireland) (eu-west-1)
- **EUC1:** EU (Frankfurt) (eu-central-1)
- **EUW2:** EU (London) (eu-west-2)
- **EUW3:** EU (Paris) (eu-west-3)
- **EUN1:** EU (Stockholm) (eu-north-1)
- **USE1:** US East (N. Virginia) (us-east-1)
- **USE2:** US East (Ohio) (us-east-2)
- **USW2:** US West (Oregon) (us-west-2)

Frequently Asked Questions in Lightsail

This topic answers frequently asked questions (FAQ). If you have a FAQ that is not answered here, use the **Questions? Comments?** feedback button at the bottom of the page. You can also post a question in the [Lightsail discussion forum](#).

Contents

- [General](#)
- [Instances](#)
- [Object storage and buckets](#)
- [Container services](#)
- [Databases](#)
- [Block storage](#)
- [Load balancers](#)
- [Content delivery network distributions](#)
- [Certificates](#)
- [Manual and automatic snapshots](#)
- [Networking](#)
- [Domains](#)
- [Billing and account management](#)
- [Export to Amazon Elastic Compute Cloud \(Amazon EC2\)](#)
- [Tags](#)
- [Contacts and notifications](#)
- [Metrics and alarms](#)

General

What is Amazon Lightsail?

Amazon Lightsail is the easiest way to get started with AWS for developers, small businesses, students, and other users who need a solution to build and host their websites and web

applications in the cloud. Lightsail provides developers compute, storage, and networking capacity. Lightsail includes everything you need to launch your project quickly – virtual machines, containers, databases, CDN, load balancers, DNS management etc. – for a low, predictable monthly price.

What can I do with Lightsail?

You can create preconfigured virtual private servers (instances) that include everything to easily deploy and manage your application, or create databases for which the security and health of the underlying infrastructure and operating system is managed by Lightsail. Lightsail is best suited to projects that require a few dozen instances or less, and developers who prefer a simple management interface. Common use cases for Lightsail include running websites, web applications, business software, blogs, e-commerce sites, and more. As your project grows, you can use load balancers and attached block storage with your instance to increase redundancy and uptime and access dozens of other AWS services to add new capabilities.

Does Lightsail offer an API?

Yes. Everything you do in the Lightsail console is backed by a publicly available API. Learn how to install and use the Lightsail [CLI](#) and [API](#).

How do I sign up for Lightsail?

To start using Lightsail, choose [Get Started](#) and log in. You use your Amazon Web Services account to access Lightsail; if you don't already have one, you'll be prompted to create one.

In which AWS Regions is Lightsail available?

Lightsail is currently available in all the Availability Zones in the following AWS Regions:

- US East (Ohio) (us-east-2)
- US East (N. Virginia) (us-east-1)
- US West (Oregon) (us-west-2)
- Asia Pacific (Mumbai) (ap-south-1)
- Asia Pacific (Seoul) (ap-northeast-2)
- Asia Pacific (Singapore) (ap-southeast-1)
- Asia Pacific (Sydney) (ap-southeast-2)

- Asia Pacific (Tokyo) (ap-northeast-1)
- Canada (Central) (ca-central-1)
- EU (Frankfurt) (eu-central-1)
- EU (Ireland) (eu-west-1)
- EU (London) (eu-west-2)
- EU (Paris) (eu-west-3)
- EU (Stockholm) (eu-north-1)

For more information, see [AWS Regions and Availability Zones in Lightsail](#).

What are Availability Zones?

Availability Zones are collections of data centers that run on physically distinct, independent infrastructure and are engineered to be highly reliable. Common points of failure such as generators and cooling equipment are not shared between Availability Zones. Additionally, Availability Zones are physically separate, so that even extremely uncommon disasters such as fires, tornadoes, or flooding can affect only a single Availability Zone.

What are the Lightsail service quotas?

For the latest Lightsail service quotas, including which quotas can be increased, see the [Lightsail service quotas](#) in the *AWS General Reference*. If you need to increase a quota, please open a case with [AWS Support](#).

How can I get more help?

We're here for you. Our context-sensitive **Help** panel in Lightsail offers immediate helpful tips about your actions in the console. From the Lightsail console, you can also access a library of [getting started guides](#), [overviews](#), and [how-to topics](#). And if you want to use the Lightsail API, or AWS CLI, Lightsail has a full API reference for all supported programming languages. You can also use Lightsail support resources.

If you have an issue with your account or billing, contact [AWS Support](#) online. You get free 24x7 access with your Lightsail account.

If you have a general question about how to use Lightsail, search the Lightsail documentation and [support forums](#).

Additionally, AWS Support offers an array of paid plans to cover your individual needs.

Instances

What is a Lightsail instance?

A Lightsail instance is a virtual private server (VPS) that lives in the AWS Cloud. Use your Lightsail instances to store your data, run your code, and build web-based applications or websites. Your instances can connect to each other and to other AWS resources through both public (internet) and private (VPC) networking. You can create, manage, and connect easily to instances right from the Lightsail console.

What is a Lightsail plan?

Also referred to as a bundle, a Lightsail plan includes a virtual server with a fixed amount of memory (RAM) and compute (vCPUs), SSD-based storage (disks), and a free data transfer allowance. Lightsail plans also offer static IPv4 addresses, and DNS management. Lightsail plans are charged on an hourly, on-demand basis, so you only pay for a plan when you're using it.

What software can I run on my instances?

Lightsail offers a range of operating system and application templates that are automatically installed when you create a new Lightsail instance. Application templates include WordPress, WordPress Multisite, cPanel & WHM, PrestaShop, Django, Drupal, Ghost, Joomla!, Magento, Redmine, LAMP, Nginx (LEMP), MEAN, and Node.js.

You can install additional software on your instances by using the in-browser SSH or your own SSH client.

What operating systems can I use with Amazon Lightsail?

Lightsail currently supports 7 Linux or Unix-like distributions: AlmaLinux OS 9, Amazon Linux 2, Amazon Linux 2023, CentOS, Debian, FreeBSD, OpenSUSE, and Ubuntu, as well as three Windows Server versions: 2016, 2019, and 2022.

Do I need to bring my own license to use Lightsail instances?

All instance blueprints available on Lightsail include a license, except for the cPanel & WHM blueprint. That blueprint includes a 15-day trial license. For more information, see [Quick start guide: cPanel & WHM on Amazon Lightsail](#). For all other instance blueprints, you don't need to bring your own license (BYOL).

How do I create a Lightsail instance?

After logging in to Lightsail, you can use the Lightsail [console](#), command line interface (CLI), or API to create and manage instances.

The first time you log in to the console, choose Create Instance. The create instance page is where you can choose the software, location, and name for your instance. Once you choose Create, your new instance will spin up automatically within minutes.

How do Lightsail instances perform?

Lightsail instances are specifically engineered by AWS for web servers, developer environments, and small database use cases. Such workloads don't use the full CPU often or consistently, but occasionally need a performance burst. Lightsail uses burstable performance instances that provide a baseline level of CPU performance with the additional ability to burst above the baseline. This design enables you to get the performance you need, when you need it, while protecting you from the variable performance or other common side effects that you might typically experience from over-subscription in other environments.

If you need highly configurable environments and instances with consistently high CPU performance for applications such as video encoding or HPC applications, we recommend you use [Amazon EC2](#).

How do I know when my instances are bursting?

On the CPU utilization metric graphs for your instance, you will see a sustainable zone, and a burstable zone. Your Lightsail instance can operate in the sustainable zone indefinitely with no impact to the operation of your system. Your instance may begin operating in the burstable zone when under heavy load. While operating in the burstable zone your instance is consuming a higher amount of CPU cycles. Therefore, it can only operate in this zone for a limited period of time. For more information, see [Viewing instance metrics in Amazon Lightsail](#).

Add a metric alarm to be notified when your instance's CPU utilization crosses from the sustainable zone to the bursting zone. For more information, see [Creating instance metric alarms in Amazon Lightsail](#).

How do I connect to a Lightsail instance?

Lightsail offers a 1-click secure connection to your instance's terminal right from your browser, supporting SSH access for Linux/Unix-based instances and RDP access for Windows-based instances. To use 1-click connections, launch your instance management screens, choose

Connect using SSH or **Connect using RDP**, and a new browser window opens and automatically connects to your instance.

If you prefer to connect to your Linux/Unix-based instance using your own client, Lightsail will do the SSH key storing and management work for you, and provide you with a secure key to use in your SSH client.

How can I back up my instances?

If you want to back up your data, you can use the Lightsail console or API to create a manual snapshot of your instance, or enable automatic snapshots to have Lightsail create daily snapshots for you. If there is a failure or bad code deployment, you can later use your instance snapshot to create a brand new instance. For more information, see [Snapshots](#).

Can I upgrade my plan?

Yes. You can use a snapshot of your instance to create a new, larger size instance. For more information, see [Snapshots](#).

How can I connect Lightsail instances to other resources in my AWS account?

You can connect your Lightsail instances to Amazon VPC resources in your AWS account privately, by using VPC peering. Just choose **Enable VPC peering** on your Lightsail account page, and Lightsail does the work for you. Once VPC peering is enabled, you can address other AWS resources in your default Amazon VPC by using their private IPs. Find instructions [here](#).

Note

Note that you need to have a default Amazon VPC set up in your AWS account in order for VPC peering with Lightsail to work. AWS accounts created before December 2013 do not have a default VPC, and you will need to set one up. Find out more about setting up your default VPC [here](#).

What is the difference between stopping and deleting my instance?

When you stop your instance, it is powered down at its current state and is available for you to start again at any time. Stopping your instance will release its public IPv4 address, so it is recommended that you use static IPv4 addresses for instances that must retain the same IP after they are stopped and started. Note that the public IPv6 addresses attached to instances don't change even when instances are stopped and started.

When you delete your instance, you are performing a destructive action. Unless you have created an instance snapshot, all of your instance data will be lost and you cannot recover it again. Automatic snapshots are also deleted with the instance unless you keep them by copying them as manual snapshots. The instance's public and private IP addresses will also be released. If you were using a static IPv4 address with that instance, the static IPv4 address is detached, but remains in your account.

Object storage and buckets

What can I do with Lightsail object storage?

You can store your static content, such as images, videos, and HTML files in a bucket in the Lightsail object storage service. You can use the objects stored in your bucket with your websites and applications. Lightsail object storage can be associated to your Lightsail CDN distribution with a few simple clicks, making it quick and easy to accelerate the delivery of your content to a global audience. It can also be used as a low cost, secure backup solution. For more information, see [Object storage](#).

What does Lightsail object storage cost?

Lightsail object storage has three different fixed-priced bundles in all AWS Regions where Lightsail is available. The first bundle is \$1/month and is free for the first 12 months. This bundle includes 5 GB storage capacity and 25 GB of data transfer. The second bundle is \$3 per month and includes 100 GB storage capacity and 250 GB of data transfer. Lastly, the third bundle is \$5 per month and includes 250 GB of storage capacity and 500 GB data transfer. Lightsail object storage includes unlimited data transfer into your bucket, as the bundled data transfer allowance is used only for data transfer out from your bucket.

Does Lightsail object storage have overage charges?

When you exceed the monthly storage capacity or data transfer allowance of the selected storage plan for an individual bucket, you will get charged for the additional amount. For more information, see the [Lightsail pricing page](#).

How does my data transfer allowance work with object storage?

You can consume your data transfer allowance by transferring data into and out of Lightsail object storage, except for the following:

- Data transferred into Lightsail object storage from the internet

- Data transfer between Lightsail object storage resources
- Data transferred out from Lightsail object storage to another Lightsail resource in the same AWS Region (including to a resource in a different AWS account, but in the same AWS Region)
- Data transferred out from Lightsail object storage to a Lightsail CDN distribution

Can I change the plan associated with my Lightsail bucket?

Yes, you can change the storage plan of an individual Lightsail bucket one time within your monthly AWS billing cycle.

Can I copy objects from Lightsail object storage to Amazon S3?

Yes, copying from Lightsail object storage to Amazon S3 is supported. For more information, see [How can I copy all objects from one Amazon S3 bucket to another bucket?](#) in the *AWS Premium Support Knowledge Center*.

How do I get started with Lightsail object storage?

To use Lightsail object storage, you must first create a bucket that is used to store your data. For more information, see [Create a bucket](#). After your bucket is up and running, you can start adding objects to your bucket by uploading files using the Lightsail console or by configuring your application to put content like logs or other application data in the bucket. Alternatively, you can also get started with Lightsail object storage through the use of AWS Command Line Interface (AWS CLI).

How do I upload objects to my bucket?

To upload object(s) to your bucket, like images or other static files, choose “Upload” from the “Objects” top navigation tab and select the correct file or directory from your computer. Alternately, drag and drop files and directories from your desktop into the marked area in the Lightsail object storage console.

Can I block public access to my bucket?

Lightsail buckets and objects are set to private by default, meaning that only users with appropriate permissions have access to the bucket and objects. A user can change this default setting and either make individual objects public and read only in a private bucket or opt to make the entire bucket public and read only. When a user makes a bucket or object public, anyone in the world can read its content. For more information, see [Bucket permissions](#).

How do I provide programmatic access to my bucket?

You can use either access keys or roles for programmatic access to your bucket. First, select the bucket you want to programmatically connect to in the Lightsail console. Second, under the **Permissions** tab, create an access key or assign a role to your Lightsail instance and then configure your website or application code to use your bucket. This behavior may vary depending on how you plan to use object storage with your website or application. For more information, see [Bucket permissions](#).

How do I share a bucket with other AWS accounts?

Lightsail makes cross-account sharing easy by allowing you to share access to your bucket with the AWS account ID that you specify in the Cross-account access section of the bucket management page. After you specify an AWS account ID, that account will have read-only access to the bucket. For more information, see [Bucket permissions](#).

What is versioning?

Versioning allows you to preserve, retrieve, and restore every version of every object storage in your bucket, providing an additional level of protection from accidental overwrites and deletes.. For more information, see [Enable and suspend bucket object versioning](#).

How do I associate my Lightsail bucket to my Lightsail CDN distribution?

Lightsail object storage can be associated to Lightsail CDN distributions with a few simple clicks, making it quick and easy to accelerate the delivery of your content to a global audience. To do so, create a Lightsail CDN distribution and simply select the Lightsail bucket as the origin of your Lightsail CDN distribution. For more information, see [Using an Amazon Lightsail bucket with a Lightsail content delivery network distribution](#).

What limits are there for the Lightsail object storage service?

You can create up to 20 buckets in the Lightsail object storage service per account. There is no limit to the number of objects that you can store in a bucket. You can store all of your objects in a single bucket, or you can organize them across several buckets.

Does Lightsail object storage support monitoring and alerting?

With Lightsail object storage, customers can easily view metrics on the total used space within a bucket and number of objects within the bucket. Alerting based on these metrics is also supported. For more information, see [Viewing metrics for your bucket in Amazon Lightsail](#) and [Create bucket metric alarms](#).

Container services

What can I do with Lightsail container services?

Lightsail container services provide an easy way to run containerized applications in the cloud. You can run a variety of applications on a container service, ranging from simple web apps to multi-tiered micro services. You just specify the container image, power (CPU, RAM) and scale (number of nodes) required for your container service. Lightsail takes care of running the container service without you having to manage any underlying infrastructure. Lightsail will provide you with a load balanced TLS endpoint to access the application running on the container service.

Can Lightsail container service run Docker containers?

Yes. Lightsail supports Linux-based Docker containers. Windows containers are currently not supported.

How do I use my public container images with Lightsail container service?

You can use container images from an online public registry, such as Amazon ECR Public Registry, or build your own custom image and push it to Lightsail in a few easy steps using the AWS CLI. For more information, see [Push and manage container images](#).

Can I pull my container images from a private container registry?

Currently, only public container registries are supported by Lightsail container services. Alternately, you can push your custom container images from your local machine to Lightsail to keep them private.

Can I change the power and scale of my service based on demand?

Yes, container service power and scale can be changed at any time even after the service is created.

Can I customize the name of the HTTPS endpoint created by Lightsail container service?

Lightsail provides a HTTPS endpoint for every container service in the format `<service-name>.<random-guid>.<aws-region-name>.cs.amazonlightsail.com`. Only the service name can be customized. Alternately, you can use a custom domain name. For more information, see [Enable and manage custom domains](#).

Can I use custom domains for the HTTPS endpoint of a Lightsail container service?

Yes. You can create and attach an SSL/TLS certificate with custom domain names to your container service in Lightsail. The certificates must be domain validated. If the DNS of your domain uses a Lightsail DNS zone, you can route traffic for the apex of your domain (example.com) or a subdomain (www.example.com) to your container services. Alternately, you can use a DNS hosting provider who supports adding ALIAS records to map the apex of your domain (example.com) to the default domain (public DNS) of your Lightsail container service. For more information, see [Enable and manage custom domains](#).

What do Lightsail container services cost?

Lightsail container services are billed on an on-demand hourly rate, so you pay only for what you use. For every Lightsail container service you use, we charge you the fixed hourly price, up to the maximum monthly service price. Maximum monthly service price can be calculated by multiplying the base price of the power of your service with the scale of your service. For example, a service of Micro power and scale of 2 will cost a maximum of $\$10 \times 2 = \20 /month. The least expensive Lightsail container service starts at \$0.0094 USD/hour (\$7 USD/month). Additional data transfer charges may apply for usage above the free-quota of 500 GB per month for each service.

Will I be charged for the whole month even if I run my container service for a few days?

Your Lightsail container services are charged only when they're in the running or disabled state. If you delete your Lightsail container service before the end of the month, we charge you a prorated cost based on the total number of hours that you used your Lightsail container service. For example, if you use your Lightsail container service with a power of Micro and scale of 1 for 100 hours in a month, you will be charged \$1.34 ($\0.0134×100)

Will I be charged for data transfer in and out of the container service?

Every container service comes with a data transfer quota (500 GB per month). This counts towards both the data transfer IN and OUT of your service. When you exceed the quota, you will get charged for data transfer OUT from a Lightsail container service to the Internet or to another AWS Region or to AWS resources in the same Region when using public IP addresses. The charge for these types of data transfer above the free allowance is as follows:

- US East (Ohio) (us-east-2): \$0.09 USD/GB
- US East (N. Virginia) (us-east-1): \$0.09 USD/GB
- US West (Oregon) (us-west-2): \$0.09 USD/GB

- Asia Pacific (Mumbai) (ap-south-1): \$0.13 USD/GB
- Asia Pacific (Seoul) (ap-northeast-2): \$0.13 USD/GB
- Asia Pacific (Singapore) (ap-southeast-1): \$0.12 USD/GB
- Asia Pacific (Sydney) (ap-southeast-2): \$0.17 USD/GB
- Asia Pacific (Tokyo) (ap-northeast-1): \$0.14 USD/GB
- Canada (Central) (ca-central-1): \$0.09 USD/GB
- EU (Frankfurt) (eu-central-1): \$0.09 USD/GB
- EU (Ireland) (eu-west-1): \$0.09 USD/GB
- EU (London) (eu-west-2): \$0.09 USD/GB
- EU (Paris) (eu-west-3): \$0.09 USD/GB
- EU (Stockholm) (eu-north-1): \$0.09 USD/GB

What is the difference between stopping and deleting my container service?

When you disable your container service, your container nodes are in a disabled state and the public endpoint of the service returns a HTTP status code '503'. Enabling the service restores it to the last active deployment. Power and scale configurations are also retained. Public endpoint name does not change after re-enabling. Deployment history and container images are preserved.

When you delete your container service, you are performing a destructive action. All the container nodes of the service will be permanently deleted. The HTTPS public endpoint address, container images, deployment history, and logs associated with your service will also be permanently deleted. You will not be able to recover the endpoint address.

Will I be charged if my container service is in a disabled state?

Yes, you are charged according to the power and scale configuration of your container service, even when it is in a disabled state.

Can I use container services as the origin to my Lightsail content delivery network (CDN) distributions?

Container services are currently not supported as origins for Lightsail CDN distributions.

Can I use container services as targets for my Lightsail load balancer?

No. Container services are currently not available as targets for Lightsail load balancers. However, the public endpoints of container services come with built-in load balancing.

Can I configure the public endpoint of my container service to redirect HTTP requests to HTTPS?

Lightsail container service public endpoints automatically redirect all HTTP requests to HTTPS to ensure that your content is served securely.

Do container services support monitoring and alerting?

Container services provide metrics for CPU utilization and memory utilization across the nodes of your service. Alerting based on these metrics is currently not supported.

Do Lightsail container services support IPv6?

Lightsail container service HTTPS endpoints support both IPv4 and IPv6. Pv6 cannot be disabled on container services.

Databases

What are Lightsail managed databases?

Lightsail managed databases are instances that are dedicated to running databases, instead of other workloads like web servers, mail servers, etc. A managed database can contain multiple user-created databases, and you can access it by using the same tools and applications that you use with a stand-alone database. Lightsail maintains the security and health of your database's underlying infrastructure and operating system, so that you can run a database without deep expertise in infrastructure management.

Like regular Lightsail instances, Lightsail managed databases come with a fixed amount of memory, computing power, and SSD based storage in their plans that you can scale up over time. Lightsail will automatically install and configure your chosen database for you upon creation.

What can I do with Lightsail managed databases?

Lightsail managed databases provide an easy, low maintenance way to store your data in the cloud. You can run managed databases either as a new database or by migrating from an existing on-premises or hosted database to Lightsail.

They can also allow you to scale your application to accept larger amounts of traffic and more intensive loads, by separating out your database into a dedicated instance. Lightsail managed databases are especially useful for stateful applications – like WordPress and most common CMSs – that need data to be kept in sync when you scale beyond a single instance. Managed databases can be paired with a Lightsail load balancer and two or more Lightsail instances to create a powerful, scaled application. By using Lightsail high availability managed database plans, you can also add redundancy to your database, helping to ensure high uptime for your application.

What does Lightsail manage for me?

Lightsail manages a range of maintenance activities and security for your managed database and its underlying infrastructure. Lightsail automatically backs up your database and allows point in time restore from the past 7 days using the database restore tool, to help protect against data loss or component failure. Lightsail also automatically encrypts your data at rest and in motion for increased security and stores your database password for easy and secure connections to your database. On the maintenance side, Lightsail runs maintenance on your database during your set maintenance window. This maintenance include automatic upgrades to the latest minor database version and all management of the underlying infrastructure and operating system.

What kinds of databases and what versions of these databases does Lightsail support?

Lightsail managed databases support the latest major versions of MySQL and PostgreSQL. Currently, these versions are MySQL 5.7, MySQL 8.0, PostgreSQL 9, PostgreSQL 10, PostgreSQL 11, and PostgreSQL 12. Lightsail only provides the latest minor version for each major version option.

What managed database plans does Lightsail offer?

Lightsail offers 4 sizes of managed databases in standard and high availability plans. Each plan comes with a fixed amount of storage and a monthly allowance of data transfer. You can also scale up to larger plans over time, as needed, and switch between standard and high availability plans. High availability plans mirror the same resources as standard plans and additionally include a standby database running in a separate Availability Zones from your primary database for redundancy.

What is a high availability plan?

Lightsail managed databases are available in standard and high availability plans. Standard and high availability plans have identical plan resources, including memory, storage, and data transfer allowance. High availability plans add redundancy and durability to your database,

by automatically creating standby database in a separate Availability Zone from your primary database, synchronously replicating data to the standby database, and providing failover to the standby database in case of infrastructure failure and during maintenance so that you ensure uptime even when databases is being automatically upgraded/maintained by Lightsail. Use high availability plans for running production applications or software where high uptime is required.

How do I scale up or down my Lightsail managed database?

You can scale up your Lightsail managed database by taking a snapshot of it and creating a new, larger database plan from snapshot or by creating a new, larger database using the emergency restore feature. You can also switch from standard to high availability plans and vice versa using either method. You cannot scale down your database. For more information, see [Creating a database from a snapshot in Amazon Lightsail](#).

How can I back up my Lightsail managed database?

Lightsail backs up your data automatically and allows restore of this data from a specific point in time to a new database. Automatic backup is a free service for your database but only saves the last 7 days of data. If you delete your database, all automatic backup records are deleted and point-in-time restore is no longer possible. To retain backups of data after deleting your database or to retain a backup for more than 7 days in the past, use manual snapshots.

You can take manual snapshots of your Lightsail managed databases from the database management pages. Manual snapshots contain all the data from your database and can be used as backups for data that you want to store permanently. You can also use manual snapshots to create a new, larger database or to switch between Standard and High Availability plans. Manual snapshots are stored until you delete them and are billed at \$0.05 USD/GB-month.

What happens to my data if I delete my Lightsail managed database?

If you delete your Lightsail managed database, both your database itself and all automatic backups will be deleted. There is no way to recover this data unless you take a manual snapshot before deleting your database. During deletion of your database, Lightsail provides a one-click option to take a manual snapshot, if desired, to help protect against accidental loss of data. Taking a manual snapshot before deletion is optional but highly recommended. You can delete your manual snapshot in the future when you no longer need the stored data.

Can I connect my instance(s) to a Lightsail managed database running in different AWS Regions or different Availability Zones?

You cannot use Lightsail managed databases with instances running in different AWS Regions. You can, however, use databases across different Availability Zones from your instance.

How do I load data onto my Lightsail managed database?

In order to load data onto your Lightsail managed database, you should first enable data import mode. After enabling data import mode, you can continue to manually upload data using your preferred database client. After you are done loading data, remember to turn off Data import mode so that automatic backups and logging for your databases can resume. For more information, see [Import data into your MySQL database](#) and [Import data into your PostgreSQL database](#).

How do I access the data on my Lightsail managed database?

You can connect to your database and query your data using any standard SQL client application. We recommend MySQL Workbench for GUI based administration and querying. You can find connection data in the database management screen for your database, including the endpoint URL and DNS name. For more information, see [Connect to your MySQL database](#) or [Connecting to your PostgreSQL database in Amazon Lightsail](#).

How do Lightsail managed databases work with my Lightsail instances?

After you create your Lightsail managed database, you can start using it with your application immediately, using your Lightsail instances as web servers or other dedicated workloads for your app. To connect your Lightsail instance to a database, use your database endpoint and reference your securely stored password to configure the database as your data store in the code of your application. You can find connection data in the database management screens. The file name and location for your database configuration file will vary by application. Note that you can connect many instances to one database, either using the same tables or using different ones.

How can I connect Lightsail managed database to EC2 instances running in my AWS account?

You can connect your Lightsail managed database to EC2 instances by connecting over the public internet. Note that connection to all AWS services will consume your database data transfer allowance, and data out over the public internet to AWS services in excess of your data transfer allowance will accrue overage charges. You cannot use VPC peering between Lightsail managed databases and EC2 instances.

What is the difference between public and private modes for my Lightsail managed database?

By default, your Lightsail managed database is created in private mode, which secures it by making it accessible only by Lightsail instances. You can set your database public mode if you need to connect to software or services over the public internet. To ensure security of your data,

we do not recommend keeping public mode enabled long-term. You can change between public and private modes at any time from your database management screens.

Can I manage the ports used by my Lightsail managed database?

No, Lightsail automatically manages your ports for security purposes, opening Port 3306 for MySQL for all Lightsail managed databases in public mode. If your database is in private mode, your database is only open to resources running in your Lightsail account via the internal network.

Do Lightsail managed databases services support IPv6?

Lightsail managed databases do not support IPv6.

Block storage

What can I do with Lightsail block storage?

Lightsail block storage provides additional storage volumes (called "attached disks" in Lightsail) that you can attach to your Lightsail instance, similar to an individual hard drive. Attached disks are useful for applications or software that need to separate out specific data from their core service and to protect application data in case of a failure or other issue with your instance and system disk. Attached disks offers consistent performance and low latency needed for applications or software that frequently access their stored data.

Lightsail block storage disks use solid-state drives (SSD). This type of block storage balances a low price and good performance and is intended to support the vast majority of workloads that run on Lightsail. For customers with applications that require sustained IOPS performance, high amounts of throughput per disk, or that are running large databases like MongoDB, Cassandra, etc., we recommend using Amazon EC2 with GP2 or Provisioned IOPS SSD storage instead of Lightsail.

How are attached disks different than the storage included in my Lightsail plan?

The system disk included with your Lightsail plan is your instance's root device. If you terminate your instance, the system disk will be deleted as well. If you experience an instance failure, the system disk could be impacted. You also cannot detach your system disk or back it up separately from your instance. Data stored on an attached disk persists independently of the instance. Attached disks can be detached and moved between instances. They can be backed up independently from an instance by creating a manual snapshot of the disk. To protect your

data, we recommend that you use your Lightsail instance's system disk only for temporary data. For data requiring a higher level of durability, we recommend using attached disks and regularly backing up your disk using disk or instance snapshots.

How large can I make my attached disk?

Each attached disk can be up to 16 TB, and the total amount of attached block storage in a Lightsail account must not exceed 20 TB.

How many disks can I attach per Lightsail instance?

You can attach up to 15 disks to a Lightsail instance.

Can I attach a disk to more than one instance?

No, disks can only be attached to one instance at a time.

Does my disk need to be attached to an instance?

No, you can choose not to attach a disk to an instance. The disk will remain in your account in an unattached state. There is no difference in price if your disk is not attached to an instance.

Can I increase the size of my attached disk?

Yes, you can increase the size of a disk by taking a disk snapshot and then creating a new, larger disk from that snapshot.

Does Lightsail block storage offer encryption?

Yes, to help keep your data secure, all Lightsail attached disks and disk snapshots are encrypted at rest by default, using keys that Lightsail manages on your behalf. Lightsail also provides encryption of data as it moves between Lightsail instances and attached disks.

What availability can I expect from Lightsail block storage?

Lightsail block storage is designed to be highly available and reliable. Each attached disk is automatically replicated within its Availability Zone to protect you from component failure. Lightsail block storage disks are designed for 99.99% availability. Lightsail also supports disk snapshots to allow regular backups of your data.

How do I back up my attached disk?

You can back up your disk by creating a manual snapshot of the disk. You can also back up your entire instance and any attached disks by creating a manual snapshot of the instance, or by enabling automatic snapshots for the instance with the disk attached. Disks attached to instances are included in instance manual and automatic snapshots.

Load balancers

What can I do with Lightsail load balancers?

Lightsail load balancers allow you to build highly available websites and applications. By distributing traffic across instances in different Availability Zones and pointing traffic to only healthy target instances, Lightsail load balancers reduce the risk of your application going down due to an issue with your instance or to a datacenter outage. With Lightsail load balancers and multiple target instances, your website or application can also accommodate increases in web traffic and maintain good performance for your visitors during peak load times.

In addition, you can use Lightsail load balancers to help you build secure applications and accept HTTPS traffic. Lightsail takes the complexity out of requesting, provisioning, and maintaining SSL/TLS certificates. The built-in certificate management requests and renews certificates on your behalf and adds the certificate to your load balancer automatically.

Can I use load balancers with instances in different AWS Regions or different Availability Zones?

You cannot use load balancers with instances running in different AWS Regions. You can, however, use target instances across different Availability Zones with your load balancer. In fact, we recommend that you distribute your target instances across Availability Zones to maximize the availability of your application.

How does my Lightsail load balancer deal with traffic spikes?

Lightsail load balancers scale automatically to handle traffic spikes to your application without you having to manually adjust them. If your application experiences a transient spike in traffic, your Lightsail load balancer will automatically scale and continue to efficiently direct traffic to your Lightsail instances. While your Lightsail load balancer is designed to easily manage traffic spikes, applications that consistently experience very high volume levels of traffic may experience performance degradation or throttling. If you expect your application consistently to manage more than 5 GB/hour of data or consistently to have a large number of connections (>400k new connections/hour, >15k active, concurrent connections), we recommend using Amazon EC2 with Application Load Balancing instead.

How do Lightsail load balancers route traffic to my target instances?

Lightsail load balancers direct traffic to your healthy target instances based on a round robin algorithm.

How does Lightsail know if my target instances are healthy?

After you create your load balancer and attach your instances, Lightsail sends a health check request to the root of your web application. You can customize the location by specifying a path (a common file or webpage URL) for Lightsail to ping. If the target instance can be reached using this path, then Lightsail will route traffic there. If one of your target instances is unresponsive, the health check fails and Lightsail will not route traffic to that instance. [Learn more about health checking](#)

How many instances I can attach to my load balancer?

You can add as many target instances to your load balancer as you would like - up to your Lightsail account instance quota.

Can I assign one instance to multiple load balancers?

Yes, Lightsail supports adding instances as target instances for more than one load balancer, if desired.

What happens to my target instances when I delete my load balancer?

If you delete your load balancer, the attached target instances will continue to run normally and will appear in the Lightsail console as regular Lightsail instances. Please note that you will likely need to update your DNS records to direct traffic to one of your former target instances after you delete the load balancer.

What is session persistence?

Session persistence enables the load balancer to bind a visitor's session to a specific target instance. This ensures that all requests from the user during the session are sent to the same target instance. Lightsail supports session persistence for applications that require visitors to hit the same target instances for data consistency. For example, many applications that require user authentication can benefit from using session persistence. You can turn on session persistence for specific load balancer from the load balancer management screens after creation. For more information, see [Enable session persistence for a load balancer](#).

What kind of connections do Lightsail load balancers support?

Lightsail load balancers support HTTP and HTTPS connections.

Do Lightsail load balancers support IPv6?

Lightsail load balancers created after January 12, 2021, operate in dual-stack mode by default (i.e., they accept client traffic over both IPv4 and IPv6 protocol). IPv6 can be enabled on

load balancers created before this date through a toggle on the Networking tab of the load balancer's management page. IPv6 can be disabled on any load balancer using this toggle too.

Do the instances behind a load balancer need to be IPv6 enabled to use the load balancer which is IPv6 enabled?

No. Load balancers accept both IPv4 and IPv6 traffic, and seamlessly convert it to IPv4 when communicating with the instances in the backend. Hence, instances behind a load balancer can either be dual-stack or IPv4 only.

Content delivery network distributions

What can I do with Lightsail CDN distributions?

Lightsail content delivery network (CDN) distributions make it easy for you to accelerate the delivery of content hosted on your Lightsail resources by storing and serving it on Amazon's global delivery network, powered by Amazon CloudFront. Distributions also help you enable your website to support HTTPS traffic by providing simple SSL certificate creation and hosting. Finally, distributions can help reduce the load on your Lightsail resources and help your website handle large traffic spikes. Like all of Lightsail's features, setup can be completed with just a few clicks, and you pay a simple monthly price.

What types of resources can I use as the origin of my distributions?

Lightsail distributions allow you to use your Lightsail instances and load balancers as origins. Lightsail containers are not currently supported as origins. Resources outside of Lightsail, such as S3 buckets, are not supported.

Do I need to attach a static IPv4 address to my Lightsail instance in order to use it as an origin for my Lightsail distribution?

Yes, static IPv4 addresses are required to be attached to instances that are specified as origins. Lightsail distributions do not currently support IPv6.

How do I setup a Lightsail distribution with my WordPress website?

Create your distribution, select your WordPress instance as the origin, choose your plan, and you're all set. Lightsail distributions automatically configure your distribution settings to optimize performance for most WordPress configurations.

Can I attach multiple origins?

Although you cannot attach multiple origins to your Lightsail distribution, you can attach multiple instances to a Lightsail load balancer and specify it as the origin of your distribution.

Do Lightsail distributions support certificate creation?

Yes. Lightsail distributions makes it easy to create, verify, and attach certificates directly from your distribution's management page.

Is a certificate required?

A certificate is only required if you wish to use your custom domain name with your distribution. All Lightsail distributions are created with a unique Amazon CloudFront domain name that is HTTPS-enabled. However, if you wish to use your custom domain with your distribution, then you need to attach a certificate for your custom domain to your distribution.

Is there a limit to the number of certificates I can create?

Yes, refer to the [Lightsail service quotas](#) for more information.

How can I configure my distribution to redirect HTTP requests to HTTPS?

Lightsail distributions automatically redirect all HTTP requests to HTTPS to ensure that your content is served securely.

How can I configure my apex domain to point to my Lightsail distribution?

In order to point your apex domain to your CDN distribution, you must create an ALIAS record in the domain name system (DNS) of your domain that maps your apex domain to your distribution's default domain. If your DNS hosting provider does not support ALIAS records, you can use Lightsail DNS zones to easily configure your apex domain to point to your distribution's domain.

What are the differences between Lightsail's instance data transfer quotas and distribution data transfer quotas?

While data transfer IN and OUT count toward your instance's data transfer quota, only data transfer OUT to your origin and to your viewers counts toward you distribution's quota. In addition, all data transfer OUT in excess of your distribution's quota is charged an overage fee, whereas some types of data transfer OUT are free for instances. Finally, Lightsail distributions use a different regional overage model, though the majority of the rates are the same as those charged for instance overage.

Can I change the plan associated with my distribution?

Yes, you can change your distribution's plan once per month. If you wish to change your plan a second time, you must wait until the beginning of the following month to do so.

How do I know if my distribution is working?

Lightsail distributions provide you with a variety of metrics that track the performance of your distribution, including the total number of requests your distribution has received, the amount of data your distribution has sent to clients and to your origin, and the percentage of requests that have resulted in errors. Additionally, you can create alerts that are linked to distribution metrics.

Can I delete cached content on my Lightsail distribution?

You can delete all cached content, but not specific files or folders.

When should I use Lightsail distributions versus Amazon CloudFront distributions?

Lightsail distributions are designed specifically for users who are hosting websites or web applications on Lightsail resources, such as instances and load balancers. If you're using another service in AWS to host your website or app, have complex configuration needs, or have a workload that involves a high number of requests per second or large amount of video streaming, we recommend that you use Amazon CloudFront.

Can I move my Lightsail content delivery network (CDN) distribution to Amazon CloudFront?

Yes, you can move your Lightsail distribution by creating a similarly configured distribution in Amazon CloudFront. All of the settings that can be configured in a Lightsail distribution can also be configured in a CloudFront distribution. Complete the following steps to move your distribution to CloudFront:

- Take a snapshot of your Lightsail instance that is configured as your distribution's origin. Export the snapshot to Amazon EC2, and then create a new instance from the snapshot in EC2. For more information, see [Export snapshots to Amazon EC2](#).

Note

Create an application load balancer in Elastic Load Balancing if you need to load balance your website or web application. For more information, see the [Elastic Load Balancing User Guide](#).

- Disable custom domains for your Lightsail distribution to detach certificates that you might have attached to it. For more information, see [Disabling custom domains for your Amazon Lightsail distributions](#).
- Using the AWS Command Line Interface (AWS CLI), run the `get-distributions` command to get a list of your Lightsail distribution's settings. For more information, see [get-distributions](#) in the *AWS CLI Reference*.
- Sign in to the [CloudFront console](#) and create a distribution with the same configuration settings as your Lightsail distribution. For more information, see [Creating a Distribution](#) in the *Amazon CloudFront Developer Guide*.
- Create a certificate in AWS Certificate Manager (ACM) that you will attach to your CloudFront distribution. For more information, see [Request a Public Certificate](#) in the *ACM User Guide*.
- Update your CloudFront distribution to use the ACM certificate you created. For more information, see [Updating your CloudFront distribution](#) in the *CloudFront User Guide*.

How is Lightsail CDN intended to be used?

Lightsail CDN distributions are created using fixed-priced bundles of data transfer to make the cost of using the service simple and predictable. Distribution bundles are designed to cover a month's worth of usage. Using distribution bundles in a way to avoid incurring overage fees (including, but not limited to, frequently upgrading or downgrading bundles, or using an excessively large number of distributions with a single origin) is beyond the intended scope of use and is not permitted. In addition, workloads that involve a high number of requests per second or large amount of video streaming are not permitted. Engaging in these behaviors may result in throttling or suspension of your data services or account.

Do Lightsail CDN distributions support IPv6?

All Lightsail CDN distributions have IPv6 enabled by default. The distribution hostnames resolve to both IPv4 and IPv6 addresses. IPv6 can be disabled by using a toggle on the Networking tab of the CDN's management page.

Do the origins need to be IPv6 enabled to work with the Lightsail CDN distributions?

No. CDN distributions accept both IPv6 and IPv4 traffic, and seamlessly convert it to IPv4 when communicating with the origins in the backend. Hence, origins behind a distribution can either be dual-stack or IPv4 only.

Certificates

How can I use Lightsail-provisioned certificates?

SSL/TLS certificates are used to establish the identity of your website or application and secure connections between browsers and your website. Lightsail provides a signed certificate to use with your load balancer, and the load balancer provides SSL/TLS termination before routing verified traffic to your target instances over the secure AWS network. Lightsail certificates can only be used with Lightsail load balancers, not with individual Lightsail instances.

How do I validate my certificate?

Lightsail certificates are domain validated, meaning that you need to provide proof of identity by validating that you own or have access to your website's domain before the certificate can be provisioned by the certificate authority. When you request a new certificate, Lightsail will attempt to automatically validate the certificate. If the certificate cannot be validated automatically, Lightsail will prompt you to add a CNAME record to the DNS zone(s) of the domain or domains you are validating. You'll have 72 hours to add the CNAME record wherever you currently manage your DNS zones – either Lightsail DNS management or an external DNS hosting provider.

What happens if I cannot validate my domain?

You must be able to validate that you own a domain for security purposes. This means if you or someone in your organization can't add a DNS record to validate your certificate for any reason, you will not be able to use an HTTPS-enabled load balancer with Lightsail.

How many domains and subdomains can I add to my certificate?

You can add up to 10 domains or subdomains per certificate. Lightsail does not currently support wild card domains.

How can I change the domains associated with my certificate?

To change the domains (add/delete) associated with your certificate, you will need to resubmit the certificate and revalidate your ownership of the domain(s). Follow the steps in the certificate management screens to regenerate your certificate and add or remove domains when prompted.

How do I renew my certificate?

Lightsail provides managed renewal for your SSL/TLS certificates. This means that Lightsail tries to renew the certificates automatically before they expire with no action required from

you. Your Lightsail certificate must be actively associated with a load balancer before it can be automatically renewed.

What happens to my certificate when I delete my load balancer?

If your load balancer is deleted, your certificate is deleted as well. If you need to use a certificate for the same domain(s) in the future, you will need to request and validate a new certificate.

Can I download my certificate provided by Lightsail?

No, Lightsail certificates are bound to your Lightsail account and cannot be removed and used outside of Lightsail.

Manual and automatic snapshots

What are snapshots?

Snapshots are point-in-time backups of instances, databases, or block storage disks. You can create a snapshot of your resources at any time, or you can enable automatic snapshots on instances and disks to have Lightsail create snapshots for you. You can use snapshots as baselines to create new resources or to back up your data. A snapshot contains all of the data that is needed to restore your resource (from the moment when the snapshot was taken). When you restore a resource by creating it from a snapshot, the new resource begins as an exact replica of the original resource that was used to create the snapshot.

You can manually take snapshots of your Lightsail instances, disks, and databases, or you can use [automatic snapshots](#) to instruct Lightsail to take daily snapshots of your instances and disks automatically. For more information, see [Snapshots](#).

What are automatic snapshots?

Automatic snapshots are a way to schedule daily snapshots of your Linux/Unix instances in Amazon Lightsail. You can pick a time of the day, and Lightsail will automatically take a snapshot for you each day at the time you chose and always keep your seven most recent automatic snapshots. Enabling snapshots is free, you only pay for the actual storage used by your snapshots.

What are the differences between manual and automatic snapshots?

Automatic snapshots cannot be tagged or exported directly to Amazon EC2. However, automatic snapshots can be copied and converted into manual snapshots. To copy an automatic

snapshot into a manual one, choose **Keep** from the automatic snapshot's context menu to copy it as a manual snapshot.

What resources support snapshots?

Manual snapshots can be created for instances, databases, and disks.

Automatic snapshots can be enabled for Linux or Unix instances using the Lightsail console, Lightsail API, or AWS CLI, and for disks using only the Lightsail API, or AWS CLI. Automatic snapshots are not currently supported for Windows instances, or managed databases.

How long can I store snapshots?

Manual snapshots are stored until you choose to delete them. For more information, see [Deleting snapshots in Amazon Lightsail](#).

Automatic snapshots are stored until they are replaced by a newer automatic snapshots. Lightsail stores the latest seven automatic snapshots before deleting the oldest one and replacing it with the newest one. However, you can keep a specific automatic snapshot by copying it as a manual snapshot. For more information, see [Keeping automatic snapshots of instances or disks in Amazon Lightsail](#). You will be billed the [snapshot storage fee](#) for the automatic snapshots stored in your account.

How are automatic snapshots enabled?

Automatic snapshots can be enabled using the Lightsail console, Lightsail API, or AWS CLI when you create a Linux or Unix instance, or later after the instance is running.

Automatic snapshots can also be enabled for disks when you create them or after they're created; however, it can only be done using the Lightsail API, or AWS CLI.

For more information, see [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).

When are automatic snapshots created?

When you enable automatic snapshots, a default time is set based on the AWS Region where the resource is located. You can change the automatic snapshot to your preferred time of day, in hourly increments. For more information, see [Changing the automatic snapshot time for instances or disks in Amazon Lightsail](#).

How many snapshots can I store?

You can store as many manual snapshots as you'd like. However, only the latest seven automatic snapshots are stored before the oldest one is replaced with the newest one.

How are snapshots billed?

You only pay for the snapshots stored on your Lightsail account. Lightsail snapshots (manual and automatic) cost \$0.05 USD/GB-month to store.

Will I lose my snapshots if I disable automatic snapshots?

No. If you disable automatic snapshots, Lightsail will stop creating a daily snapshot, and your existing automatic snapshots will be kept. When you re-enable automatic snapshots, Lightsail will resume taking daily snapshots, deleting the oldest one and replacing it with the newest one.

What should I do if I don't want an automatic snapshot to be replaced?

You can keep a specific automatic snapshot by copying it as a manual snapshot. For more information, see [Keeping automatic snapshots of instances or disks in Amazon Lightsail](#).

Can I delete an automatic snapshot?

You can delete an automatic snapshot at any time by choosing **Delete** from the automatic snapshot's context menu. For more information, see [Delete automatic instance snapshots](#).

How can I use snapshots?

Snapshots can be used as a baseline or to create new resources if something goes wrong with the original resource. Snapshots can also . For more information, see [Snapshots](#).

Snapshots can also be exported to Amazon EC2 to create new resources within that service. For more information, see [Export snapshots to Amazon EC2](#).

Networking

How do I use IPs in Lightsail?

Each Lightsail instance automatically gets a private IPv4 address, a public IPv4 address, or a public IPv6 address (IPv6 must be manually enabled for instances created before January 12, 2021). You can use the private IP to transmit data between Lightsail instances and AWS resources privately, for free. You can use the public IP to connect to your instance from the Internet, such as through a registered domain name or through an SSH or RDP connection from your local computer. You can also attach a static IPv4 address to the instance, which substitutes the public IPv4 address with an IPv4 address that doesn't change even if the instance is stopped

and started. IPv6 addresses assigned to the instance remain unchanged until the instance is deleted or the IPv6 address is manually released by disabling IPv6 on the instance.

Does Lightsail support IPv6-only instances?

Yes, Lightsail instances support dual-stack (IPv4 and IPv6) and IPv6-only configurations.

What is a static IP?

A [static IP](#) is a fixed, public IP that is dedicated to your Lightsail account. You can assign a static IPv4 address to an instance, replacing its public IPv4. If you decide to replace your instance with another one, you can reassign the static IP to the new instance. In this way, you don't have to reconfigure any external systems (like DNS records) to point to a new IP address every time you want to replace your instance. Lightsail currently supports static IPs for IPv4 only. Static IPv6 addresses are not available. However, IPv6 addresses assigned to the instance remain unchanged until the instance is deleted or the IPv6 address is manually released by disabling IPv6 on the instance.

How many static IPs can I attach to an instance?

You can attach one static IP to an instance.

What are DNS records?

DNS is a globally distributed service that translates human readable names like `www.example.com` into alphanumeric IP addresses, like `192.0.2.1` that computers use to connect to each other. With Lightsail, you can easily map your registered domain names such as `photos.example.com` to the public IPs of your Lightsail instances. In this way, when users type human readable names like `example.com` into their browsers, Lightsail automatically translates the address into the IP of the instance you want to direct your users to. Each of these translations is referred to as a DNS query.

It's important to know that to use a domain in Lightsail, you must first register it. You can register domains by using [Lightsail](#), or your preferred DNS registrar.

Can I manage firewall settings for my instance?

Yes. You can control the data traffic for your instances by using the Lightsail firewall. From the Lightsail console, you can set rules about which ports of your instance are publicly accessible for different types of traffic.

Domains

What can I do with Lightsail domains?

Lightsail domains allow you to register and manage domains for your website or application. If you have domains that are registered with other providers, you can transfer management of those domains to Lightsail. You can also point those domains to your Lightsail resources.

What top-level domains (TLDs) can I use?

Lightsail uses the same generic TLDs as Amazon Route 53. If you would like to register a geographic domain, we recommend you use the Route 53 console. Your geographic domain will be available in the Lightsail console after it has been registered using Route 53. For more information about the TLDs that Lightsail supports, see [Domains that you can register with Amazon Route 53 in the Amazon Route 53 Developer Guide](#).

Can I make Lightsail the DNS service for my existing domain?

You can transfer DNS management of a domain that you registered using another DNS service provider to Lightsail. For more information, see [Create a DNS zone to manage your domain's DNS records](#).

How do I get started with domain registration in Lightsail?

After logging in to Lightsail, you can use the [Lightsail console](#) to create and manage domains. For more information, see [Domain registration](#).

When should I register a domain in Lightsail versus Route 53?

Tasks such as registering a domain, creating DNS zones, and routing traffic for a domain to Lightsail resources are done in Lightsail. We recommend using Route 53 for advanced tasks, such as extending domain registrations, transferring domains, including traffic policies, and creating private hosted zones.

Can I transfer my domain to Lightsail?

You can transfer your domain to Route 53. After the domain transfer is complete, your domain will be available in the Lightsail console. For more information, see [Managing a Lightsail domain in Amazon Route 53](#).

What Lightsail resources can I use with domains?

After registering a domain in Lightsail, you can point your domain to a Lightsail instance, container, load balancer, static IP, or content distribution network (CDN).

Billing and account management

What do Lightsail plans cost?

Lightsail plans are billed on an on-demand hourly rate, so you pay only for what you use. For every Lightsail plan you use, we charge you the fixed hourly price, up to the maximum monthly plan cost. The least expensive Lightsail plan starts at \$0.0047 USD/hour (\$3.50 USD/month). Lightsail plans that include a Windows Server license start at \$0.01075 USD/hour (\$8 USD/month).

When am I getting charged for a plan?

Lightsail instances and managed databases incur charges until they are deleted. If you delete your Lightsail instance or managed database before the end of the month, we only charge you a prorated cost, based on the total number of hours that you used your Lightsail instance or managed database for that month. For example, if you use the least expensive Lightsail instance plan for 100 hours in a month, you will be charged 46 cents (100×0.0046).

Can I try Lightsail instances for free?

Yes! Whether you're an existing or new AWS customer, you get 750 hours of free usage of the \$3.50 USD Lightsail plan for free. You also can try Lightsail plans that include a Windows Server license for free using the \$8 USD Windows plan.

You can use your 750 hours of usage across as many instances as you like. For example, you can run a single Lightsail instance for a whole month, or 10 Lightsail instances for 75 hours. The free trial offer is only applicable to usage within the first calendar month from when you sign up to use Lightsail. If your account is linked to an organization (under AWS Organizations), only one account within the organization can benefit from the AWS Free Tier offers.

Note

As part of the AWS Free Tier, you can get started with Amazon Lightsail for free on select instance bundles. For more information, see **AWS Free Tier** on the [Amazon Lightsail Pricing page](#).

When does the Lightsail free trial start?

The Lightsail free trial benefits start when the first free trial eligible resource is launched.

The extended 90 day free trial for instances and databases is applicable only on select plans (bundles). The offer applies to new or existing AWS accounts that started using Lightsail on or after July 8, 2021. For more information, see the [Lightsail pricing page](#).

What do Lightsail managed databases cost?

Lightsail managed databases come in 4 plan sizes and start at \$15 USD per month for a 1GB RAM database instance with 40 GB of SSD storage and 100 GB data transfer allowance. High Availability plans costs two times the Standard plan prices, because they run an additional database instance and storage disk in another Availability zone for redundancy.

Can I try Lightsail managed databases for free?

Yes! New Lightsail customers get 1 month of the \$15 USD Lightsail plan free.

What does Lightsail block storage cost?

Lightsail block storage costs \$0.10 USD per GB per month.

What do Lightsail load balancers cost?

Lightsail load balancers cost \$18 USD per month.

What does certificate management cost?

Lightsail certificates and certificate management are free with use of a Lightsail load balancer.

What do Lightsail static IPv4 addresses cost?

There are no costs associated with Static IP addresses when they are attached to a Lightsail instance. Static IPs cannot be attached to IPv6-only instances. IPv4 addresses are a scarce resource and Lightsail is committed to helping to use them efficiently, so we charge a small \$0.005 USD/hour fee for static IPs not attached to an instance for more than 1 hour.

What does data transfer cost?

Your instance, database, and content delivery network (CDN) distribution plans include a data transfer allowance.

For Lightsail instances, both data transfer in and data transfer out of your instance count toward your data transfer allowance. If you exceed your data transfer allowance, you will only get charged for data transfer OUT from a Lightsail instance to the internet or to AWS resources using the public IP address of the instance. Both data transfer IN to Lightsail instances and data transfer OUT from a Lightsail instance when using the instance's private IP address are free beyond your data transfer allowance.

For Lightsail managed databases, only data transfer OUT is counted against your allowance. If you exceed your data transfer allowance, you will only get charged for data transfer OUT from a Lightsail managed database to the internet.

For Lightsail CDN distributions, all data transfer out of your distribution counts toward your allowance. All data transfer out of your distribution will incur a charge after you exceed your distribution data transfer allowance.

How does my data transfer allowance work with my load balancers?

Your load balancer does not consume your data transfer allowance. Traffic between the load balancer and the target instances or distributions is metered and counts toward your data transfer allowance for your instances or distributions, in the same way that traffic in from and out to the internet is counted toward your data transfer allowance for Lightsail instances that are not behind a load balancer. Traffic into and out of your load balancer to the internet is not calculated toward the data transfer allowance for your instances.

What if I exceed my data transfer plan allowance?

We have designed our data transfer plans so that the vast majority of our customers will be fully covered by their allowance and not incur any additional charges. If your instance exceeds its plan data transfer allowance, you will be charged an overage fee per GB of data transfer used (data transfer OUT to the internet only).

Even if your instance exceeds its plan data transfer allowance, many types of data transfer are free. Data transfer IN to Lightsail instances and databases is always free. Data transfer OUT from a Lightsail instance to another Lightsail instance, in between Lightsail instances and Lightsail managed databases, or to AWS resources in the same Region is also free if private IP addresses are used.

What types of data transfer do I get charged for?

When you exceed the monthly free data transfer allowance of your instance plan, you will get charged for data transfer OUT from a Lightsail instance to the internet or to another AWS Region or to AWS resources in the same Region when using public IP addresses. The charge for these types of data transfer above the free allowance is as follows:

- US East (Ohio) (us-east-2): \$0.09 USD/GB
- US East (N. Virginia) (us-east-1): \$0.09 USD/GB
- US West (Oregon) (us-west-2): \$0.09 USD/GB

- Asia Pacific (Mumbai) (ap-south-1): \$0.13 USD/GB
- Asia Pacific (Seoul) (ap-northeast-2): \$0.13 USD/GB
- Asia Pacific (Singapore) (ap-southeast-1): \$0.12 USD/GB
- Asia Pacific (Sydney) (ap-southeast-2): \$0.17 USD/GB
- Asia Pacific (Tokyo) (ap-northeast-1): \$0.14 USD/GB
- Canada (Central) (ca-central-1): \$0.09 USD/GB
- EU (Frankfurt) (eu-central-1): \$0.09 USD/GB
- EU (Ireland) (eu-west-1): \$0.09 USD/GB
- EU (London) (eu-west-2): \$0.09 USD/GB
- EU (Paris) (eu-west-3): \$0.09 USD/GB
- EU (Stockholm) (eu-north-1): \$0.09 USD/GB

Instances created in different Availability Zones can communicate between zones privately and for free, and are much less likely to be impaired concurrently. Availability Zones enable you to build highly available applications and websites without increasing the cost of data transfer or compromising your application's security.

When you exceed the data transfer allowance of your Lightsail CDN distribution plan, you are charged for all data transfer OUT. The charge for data transfer above your distribution's allowance is different from Lightsail instances and is as follows:

- Asia Pacific: \$0.13 USD/GB
- Canada: \$0.09 USD/GB
- Europe: \$0.09 USD/GB
- India: \$0.13 USD/GB
- Japan: \$0.14 USD/GB
- Middle East: \$0.11 USD/GB
- South Africa: \$0.11 USD/GB

- South America: \$0.11 USD/GB
- United States: \$0.09 USD/GB

How do my instance data transfer plan allowances vary by AWS Region?

All AWS Regions have the same data transfer plan allowance as listed on amazonlightsail.com and amazonlightsail.com/pricing, with the exception of the Asia Pacific (Mumbai) and the Asia Pacific (Sydney) Regions. In these two AWS Regions, the data transfer plan allowance for instances is as follows:

- \$3.50 USD/month plan: .5 TB
- \$5 USD/month plan: 1 TB
- \$10 USD/month plan: 1.5 TB
- \$20 USD/month plan: 2 TB
- \$40 USD/month plan: 2.5 TB
- \$80 USD/month plan: 3 TB
- \$160 USD/month plan: 3.5 TB

Data transfer allowances for Lightsail managed databases are the same in all regions.

How does my data transfer allowance work for instances?

Every Lightsail instance plan includes a data transfer allowance. For example, using the \$3.50 USD per month plan, your instance can send to the internet and receive from the internet up to 1 TB of data each month, at no extra charge. Your data transfer allowance resets every month, and your instance can consume it whenever it needs to within the month.

After your instance reaches its data transfer allowance for the month, data transfer out to the internet is billed starting at \$0.09 USD per GB depending on the AWS Region in which your instance is located. If you delete your instance and create another one in the same month, in the same AWS Region, the free data transfer allowance is shared between the two instances.

What do Lightsail domains cost?

The prices listed in the linked .pdf file apply for new domain name registrations, renewals of existing domain name registrations as of December 22 2021. All prices include a DNS zone and

privacy protection. For information about the cost of registering domains, see [Amazon Route 53 Pricing for Domain Registration](#), and [Domain registration](#).

What does Lightsail DNS management cost?

DNS management is free within Lightsail. You can create up to 6 DNS zones and as many records as you want for each DNS zone. You also get a monthly allowance of 3 million DNS queries per month to your zones. Beyond your first 3 million queries in a month, you are charged \$0.40 USD per 1 million DNS queries.

What do Lightsail snapshots cost?

Lightsail snapshots (manual and automatic) cost \$0.05 USD/GB-month to store. This means that if you create a snapshot of an instance that is using 28 GB of space, and keep it for a month, you pay \$1.40 USD for the month.

When you take multiple, successive snapshots of the same instance, Lightsail automatically cost-optimizes your snapshots. For each new snapshot you take, you're charged only for the part of the data that changed. In the example above, if your instance data only changes by 2 GB, your second instance snapshot costs only \$0.10 USD per month.

How can I manage my AWS account?

Lightsail is an AWS service and runs on the AWS trusted and proven cloud infrastructure. You use the same AWS account and credentials to log in to Lightsail and the AWS Management Console.

You can manage your AWS account, including changing your AWS account password, user name, contact information, or billing information from the [AWS Billing and Cost Management console](#).

What are the Lightsail legal terms of use?

Lightsail is an Amazon web service, so to use Lightsail, you first agree to the [AWS Customer Agreement and Service Terms](#). When creating Lightsail instances, you also agree that your use of software is also subject to the end user license agreement of the seller, available for your review on the create instance page.

How can I pay my Lightsail bill?

You can pay and manage your bill through the AWS Billing and Cost Management console. AWS accepts most major credit cards. Learn more about managing your payment methods [here](#).

Export to Amazon Elastic Compute Cloud (Amazon EC2)

What is export to Amazon EC2?

Export to Amazon EC2 is a feature that allows you to create a copy of your Lightsail instance in Amazon EC2. When you export to Amazon EC2, you can pick among the wide set of instance types, configurations, and pricing models that Amazon EC2 offers, and have even more fine-tuned control over your networking, storage, and compute environment.

Why would I want to export to Amazon EC2?

Lightsail offers you an easy way to run and scale a wide set of cloud-based applications, at a bundled, predictable, and low price. Lightsail also automatically sets up your cloud environment configurations such as networking and access management.

Exporting to Amazon EC2 allows you to run your application on a wider set of instance types, ranging from virtual machines with more CPU power, memory, and networking capabilities, to specialized or accelerated instances with FPGAs and GPUs. In addition, Amazon EC2 performs less automatic management and set-up, allowing you more control over how you configure your cloud environment, such as your VPC.

How does exporting to Amazon EC2 work?

To get started, you need to export your manual snapshot of a Lightsail instance or block storage disk. Customers who are comfortable with Amazon EC2 can then use the Amazon EC2 creation wizard or API to create a new Amazon EC2 instances or Amazon EBS volumes, as they would from an existing EC2 AMI or EBS volume. Alternatively, Lightsail also provides a guided Lightsail console experience to help you easily create a new EC2 instance.

Note

Snapshots of cPanel & WHM, Django, and Ghost instances cannot be exported to Amazon EC2 at this time.

How am I billed?

Using the export to Amazon EC2 feature is free. Once you have exported your manual snapshots to Amazon EC2, you will be charged for the Amazon EC2 image separately and in addition to your Lightsail manual snapshot. Any new Amazon EC2 instances you launch will

also be billed by Amazon EC2, including their Amazon EBS storage volume(s) and data transfer. Refer to the [Amazon EC2 pricing page](#) for details on the pricing for your new instance and resources. Lightsail resources that continue to run in your Lightsail account will continue to be billed at their regular rates until they are deleted.

Can I export managed databases or disk snapshots?

The export feature allows you to export manual Lightsail disk snapshots but doesn't currently support manual snapshots of managed databases. Disk snapshots can be rehydrated as Amazon EBS volumes from the Amazon EC2 console or API.

What Lightsail resources can I export?

The Lightsail export to Amazon EC2 feature is designed to support the export of Linux and Windows instance snapshots to Amazon EC2. It also supports the export of block storage disk snapshots to Amazon EBS. It does not currently support the export of databases, container services, content delivery network (CDN) distributions, load balancers, static IPs, and DNS records. Additionally, snapshots of Django, Ghost, and cPanel & WHM instances cannot be exported to Amazon EC2 at this time.

Tags in Lightsail

What are tags?

A tag is a label that you assign to a Lightsail resource. Each tag consists of a key and a value, both of which you define. A tag value is optional, so you can choose to create “key-only” tags for filtering resources in the Lightsail console.

How can I use tags in Lightsail?

Tags have multiple use cases - they enable you to group and filter your resources in the Lightsail console and API, track and organize your costs in your bill, and regulate who can see or modify your resources through access management rules. By tagging your resources you can:

- **Organize** - use the Lightsail console and API filters to view and manage resources based on their tags you have assigned them. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags you've assigned to it.
- **Cost-allocate** - track and allocate costs across different projects or users by tagging your resources and creating “cost allocation tags” in the billing console. For instance, you can split out your bill and understand your costs by project or by client.

- Manage access - control how users with access to your AWS account can edit, create, and delete Lightsail resources by using AWS Identity and Access Management policies. This allows you to more easily collaborate with others without needing to give them full access to your Lightsail resources.

For more information about using tags in Lightsail, see [Tags](#).

What resources can be tagged?

Lightsail currently supports tagging for the following resources:

- Instances (Linux and Windows)
- Container services
- Block storage disks
- Load balancers
- Databases
- DNS zones
- Manual snapshots of instances, disks, and databases

Manual snapshots support tags; however, you must use the Lightsail API, or AWS CLI to tag snapshots. If you use the Lightsail console to create a manual snapshot of a tagged instance, disk, or database, the manual snapshot is automatically given the same tags as the source resource. You can edit these tags when you use the Lightsail console to create a new resource from a tagged manual snapshot.

Automatic snapshots cannot be tagged.

How can I tag my Lightsail snapshots?

The Lightsail console automatically tags manual snapshots with the same tags as its source resource. If you use the Lightsail API, or AWS CLI to create a snapshot, you can choose the tags for the snapshot yourself.

Important

Tags for manual snapshots of databases are not currently included in billing reports (cost allocation tags).

What is the difference between key-value and key-only tags?

Lightsail tags are key-value pairs, allowing you to organize resources such as instances across different categories (e.g. project:Blog, project:Game, project:Test). This allows you full control across all use cases such as resource organization, bill reporting, and access management. The Lightsail console also allows you to tag your resources with key-only tags for quick filtering in the console.

Contacts and notifications

What are notifications?

You can configure alarms in Lightsail to notify you when a metric for one of your instances, databases, or load balancers crosses a specified threshold. Notifications can be in the form of a banner displayed in the Lightsail console, an email sent to an address you specify, or an SMS text message sent to a mobile phone number you specify. To be notified by email and SMS text message, you must add your email address and mobile phone number as notification contacts in each AWS Region where you want to monitor your resources. For more information about notifications, see [Notifications](#).

How many contacts can I add?

You can add one email address and one mobile phone number in each AWS Region where you want to monitor your resources. SMS text messaging is not supported in all AWS Regions in which you can create Lightsail resources, and text messages cannot be sent to some countries and regions of the world. For more information about notifications, see [Notifications](#).

Metrics and alarms

What are metrics?

Lightsail reports metric data for instances, databases, and load balancers. Some metrics include your instance's CPU utilization percentage, the amount of inbound and outbound network traffic, system and instance error counts, database disk queue depth, database free storage space, load balancer error counts, load balancer response times, and more. Metrics allow you to monitor and maintain the reliability, availability, and performance of your resources. Monitor and collect metric data from your resources regularly so that you can more readily debug a multi-point failure, if one occurs. For more information, see [Resource metrics](#).

What are alarms?

You can create an alarm in Lightsail that watches a metric for your instances, databases, and load balancers. The alarm can be configured to notify you based on the value of the metric relative to a threshold that you specify. For more information, see [Alarms](#).

Notifications can be a banner displayed in the Lightsail console, an email sent to your email address, and an SMS text message sent to your mobile phone number. For more information about notifications, see [Notifications](#).

How many alarms can I add?

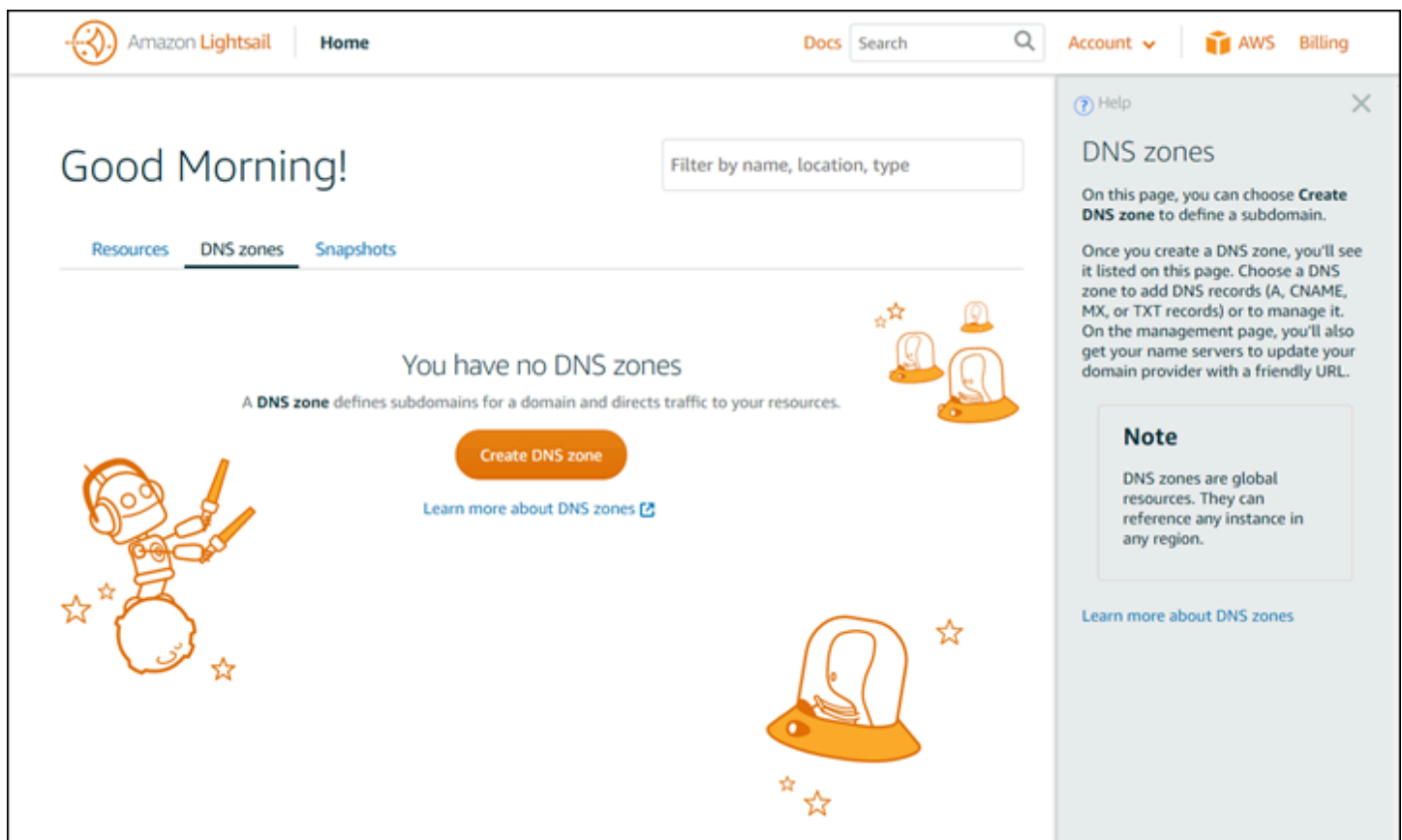
You can configure two alarms for each metric that is available for instances, databases, and load balancers. For more information, see [Alarms](#).

Get help with Amazon Lightsail

In Amazon Lightsail, you can get help in several ways.

Context-sensitive help panel

Lightsail has a context-sensitive **Help** panel on each page of the console with additional tips and information that are specific to the page you're on. Open the help panel any time you have a question about something on the page, and close it when you're good to go. You can open the help panel by choosing **Help** on any page, or by choosing any of the small question marks throughout the user interface.



About this user guide

The Amazon Lightsail user guide contains how-to topics and conceptual overviews to help you work in Lightsail. For example, you can [create an instance](#), [connect to your instance](#), or [manage your domain](#).

Using search

You can search for doc topics from any page in Lightsail by using the search box at the top of each page. To refine your search, you can search again from the documentation search page.

Didn't find what you were looking for? Sorry to hear that! Send us feedback and we'll get on it. On every page in Lightsail, you can choose **Questions?** **Comments?** and submit feedback to make suggestions. We'll get back to you with an answer.

Using the Lightsail CLI and API

You can use the AWS Command Line Interface (AWS CLI) or the Lightsail REST API to create, read, update, and delete Lightsail resources. In addition to the REST API, we also have an SDK in multiple languages, including Java, Ruby, JavaScript (Node.js), Go, PHP, Python, .NET (C#), and C++. For more information about the Lightsail API, see the [Lightsail API reference](#).

Note

You need to generate access keys to use the Lightsail API. [Learn more about setting up access keys to use the Lightsail API.](#)

The AWS CLI is helpful when you work with your Lightsail resources. In the AWS CLI, just type `aws lightsail help` to learn about the available commands. For help on a specific CLI command, type the command name followed by `help` to learn more about its parameters and exceptions. For more information, see the [Lightsail CLI reference](#).

AWS forums and other community resources

You can also post your questions in our AWS discussion forum: [AWS Forums](#).